Attachment 7

NAG3-87

DIGITAL SIMULATION OF TURBULENCE EXCITATION FOR DYNAMIC ANALYSIS OF WIND TURBINES

IN-39-CR

192953

P-118

Reza Sharif-Razi
Robert Thresher
William E. Holley

Mechanical Engineering Department
Oregon State University
Corvallis, OR 97331

March 1985

Prepared For

NATIONAL AERONAUTICS & SPACE ADMINISTRATION
Lewis Research Center
Wind Energy Project Office
21000 Brookpark Road
Cleveland, OH 44135

As a part of the

U.S. Department of Energy
Division of Distributed Solar Technology
Federal Wind Energy Program

Grant NASA NAG-3-87

DIGITAL SIMULATION OF TURBULENCE EXCITATION FOR DYNAMIC ANALYSIS OF WIND TURBINES

Reza Sharif-Razi
Robert Thresher
William E. Holley

Mechanical Engineering Department
Oregon State University
Corvallis, OR 97331

March 1985

Prepared For

NATIONAL AERONAUTICS & SPACE ADMINISTRATION
Lewis Research Center
Wind Energy Project Office
21000 Brookpark Road
Cleveland, OH 44135

As a part of the

U.S. Department of Energy
Division of Distributed Solar Technology
Federal Wind Energy Program

Grant NASA NAG-3-87

ACKNOWLEDGEMENTS

The authors would like to thank the staff of the Wind Energy Project Office of NASA Lewis Research center for their support and interest over the course of this work. In particular, Dr. David Spera has been a source of guidance not only for this small project, but for a majority of the wind activities over the past ten years. The authors are indebted to Dr. Spera for his support and guidance as is the entire Federal Wind Program.

TABLE OF CONTENTS

	Page
ABSTRACT	1
INTRODUCTION....................................	2
CHAPTER 1. TURBULENCE MODEL....................	3
1.1 Introduction.............................	3
1.2 Model Assumptions and Approximations........	4
1.3 Series Approximation to the Turbulent Velocity Field.....................................	5
1.4 Filtered Noise Model For Turbulence.........	11
1.5 References...............................	18
CHAPTER 2. NUMERICAL SIMULATION................	26
2.1 Introduction.............................	26
2.2 Generation of Uniformly Distributed Random Numbers..................................	26
2.3 Construction of a White-Noise Time Series.......	29
2.4 Filter Model.............................	31
2.5 References...............................	35
CHAPTER 3. DIGITAL COMPUTER IMPLEMENTATION.............	38
3.1 Introduction.............................	38
3.2 Input Data...............................	38
3.3 Computer Algorithm for Turbulence Simulation....	38
3.4 Tool Kit for Signal Analysis..................	41
3.5 References...............................	44

CHAPTER 4. RESULTS AND CONCLUSIONS........................ 48

4.1 Introduction.................................... 48

4.2 Comparison of Simulation 48

4.3 Concluding Remarks.............................. 51

4.4 Reference....................................... 53

APPENDIX A. LINEAR LEAST-SQUARES REGRESSION

APPENDIX B. DIGITAL SPECTRAL ANALYSIS

APPENDIX C. COMPUTER CODE LISTING

APPENDIX D. INPUT DATA FILE

APPENDIX E. PROCEDURAL EXAMPLE OF THE PROGRAM SIMULX INTERACTIVE RUN

APPENDIX F. RESULTS OF THE SAMPLE RUN FOR Mod-0A TURBINE

LIST OF FIGURES

Figure		Page
1.1	Rotor disk coordinate system.......................	24
1.2	Streamlines for in-plane velocity gradient terms...	25
2.1	Time series constructed from a sequence of uniformly distributed random numbers...............	36
2.2	Autocorrelation function and spectral density function for the constructed time series...........	37
3.1	Flow chart of the program SIMULX..................	46
3.2	Flow chart of the subroutine TURBS.................	47
4.1	Simulated series for the longitudinal velocity component V_y, as observed from the tip of a rotating MOD-0A blade with V_w = 26.25 ft/s, σ/V_w = 0.10 and L = 400 ft.........................	54
4.2	Spectral density plot of the V_y simulated time series as observed from the tip of a MOD-0A wind turbine blade.......................................	55
4.3	Probability density for the MOD-0A turbulence simulation, V_y component..........................	56
4.4	Spectral density plot of the V_z simulated time series as observed from the tip of a MOD-0A wind turbine blade.......................................	57
4.5	Probability density for the MOD-0A turbulence simulation, V_z component..........................	58
4.6	Spectral density plot of the V_y turbulence component as observed from a MOD-2 rotor blade at different radial stations.........................	59
4.7	Probability density for the V_y component of Mod-2 simulation at 70 % span..........................	60

LIST OF TABLES

Table		Page
1.1	Relative approximation error for series approximation	20
1.2	Description of turbulence input terms	21
1.3	Regression parameters for uniform turbulence terms	22
1.4	Regression parameters for shear and quadratic turbulence terms	23
3.1	List of input variables	45

LIST OF SYMBOLS

a	dimensional atmospheric coefficient
a_*	dimensionless atmospheric coefficient
A	area of rotor disk
$[A]$	12x12 diagonal atmospheric coefficient matrix containing a_* coefficients
b	dimensional atmospheric coefficient
b_*	dimensionless atmospheric coefficient
$[B]$	12x12 diagonal atmospheric coefficient matrix containing b_* coefficients
$E(\cdot)$	expectation operator
$F(\cdot)$	probability density function
f_i	weight function
$g(\cdot)$	unit impulse response
$G(\cdot)$	transfer function
i	$\sqrt{-1}$
k	the k^{th} stage or k^{th} component
L	turbulence integral scale
m	range of the random numbers
r	radial position on the rotor disk
R	rotor disk radius
$[R]$	state correlation matrix
R_f	autocorrelation function
R_*	R/L
S_w	power spectral density of white noise excitation
t	time
v_i	the turbulence velocity component

V_i	turbulence uniform term
$V_{i,j}$	turbulence gradient term
$V_{i,kj}$	turbulence quadratic term
V_w	mean wind speed
w	12x1 nondimensional vector of independent white noise
$\bar{w}$	equivalent discrete time noise input
x	12x1 vector of system states
x	lateral coordinate
x	pseudorandom number
y	longitudinal coordinate
z	vertical coordinate

Greek Symbols:

μ	expected mean value
τ	time interval
σ^2	turbulent velocity component variance
γ_{xz}	swirl about mean wind axis (in-plane)
$\bar{\gamma}_{xz}$	shear strain rate (in-plane)
ε_{xz}	shear strain rate (in-plane)
$\bar{\varepsilon}_{xz}$	dilation (in-plane)
ΔT	time interval
ϕ	transition matrix
ψ	azimuthal angular position in the rotor disk
ω	frequency in radians/sec

Mathematical Symbols:

$\triangleq$	defined equal to
$\approx$	approximately equal to

ABSTRACT

A model which approximates the three-dimensional velocity fluctuations of wind turbulence has been developed. The model provides a velocity field which varies randomly with time and space and gives the proper correlation between spatial locations and velocity components. In addition, the spectral representations approximate those observed from a rotating reference frame. The version of the model described in this report is a time domain simulation. It makes use of a random number generator to construct a white noise time series with a uniform power spectral density over the frequency range of interest. This noise source is then passed through a set of appropriate linear filters to obtain the various wind velocity fluctuations which would be experienced by a rotating wind turbine blade.

It is expected that this model of the turbulent atmospheric wind will be used as a wind simulation for other more complex dynamics codes which are used to compute dynamic loads. For this reason, the turbulence simulator has been kept as simple as possible, and was designed to compute the wind fluctuations as rapidly as possible.

INTRODUCTION

The objective of this report is to develop a time domain simulation model which approximates the three-dimensional velocity fluctuations of wind turbulence. The model provides a velocity field which varies randomly with time and space and gives the proper correlation between spatial locations and velocity components. In addition, the spectral representations approximate those observed for a rotating reference frame. It makes use of a random number generator to construct a white noise series with a uniform power spectral density over the frequency range of interest. This noise source is then passed through a set of appropriate linear filters to obtain the various wind velocity fluctuations which would be experienced by a rotating wind turbine blade.

The program is written in Fortran V on the CDC Cyber 170/720 series. It is designed in a block-structured form so various tasks performed within the program are essentially separate routines and are linked together by an executive program. Appendices C through F include a complete program listing, a sample input data file, a procedural example of the interactive features, and results of the sample run as observed from the tip of a Mod-0A wind turbine blade.

CHAPTER 1. TURBULENCE MODEL

1.1 Introduction

Fluctuations in the aerodynamic forces on a wind turbine blade are generated by the relative motions of the air with respect to the blade. These relative motions are comprised of two parts: the motions of the blade and the motions of the air. The motions of the air can further be divided into the undisturbed turbulent flow and the "induced flow" due to the presence of the wind turbine wake. The terms comprising the undisturbed turbulent flow will be characterized in this chapter. More precisely, for a horizontal axis wind turbine, the aerodynamic forces are determined by the instantaneous air velocity distribution along each of the turbine blades. These blades, in turn, are rotating through the turbulence field which is being convected past the turbine rotor disk. It is thus necessary to characterize the wind turbulence field by a three-dimensional velocity vector which varies randomly with time and with the position in space. A complete statistical description of this turbulent velocity field requires the determination of all possible joint probability distributions between different velocity components at different times and positions in space. Clearly, such a description will not be possible without considerable simplification. The validity of the resulting simplified model will depend upon a comparison of the characteristics predicted by the model and those observed in the atmosphere and more importantly, those observed in actual wind turbine field tests (1). In this chapter we will describe this model. A more detailed description of the analytical steps used to arrive at the simplified model is presented in reference (2).

1.2 Model Assumptions and Approximations

The wind turbulence inputs used in this report are determined in three basic modeling steps. First, the turbulent velocity field is characterized by a model which gives the correlations between velocity components at different spatial points and at different time instants. Second, the velocity field is approximated in the rotor disk by a series which varies with time. A correlation model for these components is derived from the original field model. Third, simple rational spectral representations are determined which approximate the derived correlation model. A brief discussion of the assumptions and approximations used in these steps follows.

The turbulent velocity field is assumed to be stationary, locally homogeneous, isotropic (3), and satisfying Taylor's frozen field hypothesis (4). The Von Karman model (5) is used to characterize the correlations between velocities of spatially separated points. This model is widely used in aircraft turbulence response analysis (6,7). However, due to the anisotropic nature of the atmospheric boundary layer, the use of the model for wind turbines can be questioned. Frost (8) has estimated that the deviation from isotropy is of secondary importance.

However, one should not rely heavily on design calculations which use this model until more complete experimental verification is available.

Once the correlation model of the turbulence field is established, the velocity is approximated over the rotor disk by a series which varies with time. This is done to simplify the statistical nature of the random field to that of several stochastic processes.

In order to further simplify the model, the power spectral densities are approximated by a simple rational form, and nondimensional parameters are determined which match the low frequency power spectral density and the total variance for the computed spectra and the rational approximation. The rational form chosen corresponds to an exponentially correlated random process which is particularly easy to handle both analytically and in simulation. The following section describes the resulting model in more detail.

1.3 Series Approximation to the Turbulent Velocity Field

The longitudinal component of turbulence (normal to the rotor disk) generally provides the most important aerodynamic effect on wind turbines (5). In order to provide an accurate determination of these effects, it is proposed to approximate the variation of the velocity across the rotor disk by a series which includes up to quadratic terms. Using Taylor's frozen field hypothesis relating the spatial and time dependency, the velocity across the rotor disk can be written as follows:

$$v_y(x, -V_w t, z) = V_{y,o}(t) + V_{y,z}(t)z + V_{y,x}(t)x$$

$$+ V_{y,zz}(t)\left(z^2 - \frac{1}{4} R^2\right) + V_{y,xx}(t)\left(x^2 - \frac{1}{4} R^2\right)$$

$$+ V_{y,zx}(t)zx \qquad (1.1)$$

where $v_y(x,y,z)$ is the velocity component depending on the x,y,z coordinates shown in Figure 1.1 and R is the radius of the rotor disk. The series of functions:

$$f_0 = 1$$

$$f_1 = z$$

$$f_2 = x \qquad (1.2)$$

$$f_3 = z^2 - \frac{1}{4} R^2$$

$$f_4 = x^2 - \frac{1}{4} R^2$$

$$f_5 = zx$$

were found by choosing polynomials with successively higher powers of x and z and enforcing conditions of mutual orthogonality over the rotor disk, i.e.,

$$\int f_j(x,z)f_k(x,z)\ dA = 0\ ;\ \text{for } j \neq k \qquad (1.3)$$

Thus, the least-square functional approximation (i.e., the terms V_y,... which minimize the difference between v_y and the approximate value) is given using the usual generalized Fourier expansion formulas (6):

$$V_{y,o} = \int (1) v_y \, dA / \int (1)^2 \, dA$$

$$V_{y,z} = \int z \, v_y \, dA / \int z^2 \, dA$$

$$V_{y,x} = \int x \, v_y \, dA / \int x^2 \, dA \tag{1.4}$$

$$V_{y,zz} = \int \left(z^2 - \frac{1}{4} R^2\right) v_y \, dA / \int \left(z^2 - \frac{1}{4} R^2\right)^2 dA$$

$$V_{y,xx} = \int \left(x^2 - \frac{1}{4} R^2\right) v_y \, dA / \int \left(x^2 - \frac{1}{4} R^2\right)^2 dA$$

$$V_{y,zx} = \int zx \, v_y \, dA / \int (zx)^2 \, dA$$

Note that the time argument has been dropped for these equations. It should be understood that these equations apply at any instant of time. Now, when the statistics of the terms $V_{y,zz}$ and $V_{y,xx}$ are considered it is found that correlation between the terms exists which complicates the statistical modeling. To alleviate this problem, linear combinations of the last three terms are defined so that the resulting six terms are all mutually uncorrelated. Thus, we define

$$V_{y,rr} = \frac{1}{2} \left(V_{y,zz} + V_{y,xx}\right)$$

$$V_{y,rc} = \frac{1}{2} \left(V_{y,zz} - V_{y,xx}\right) \tag{1.5}$$

$$V_{y,rs} = \frac{1}{2} V_{y,zx}$$

Converting to polar coordinates and substituting Eqs. (1.5) into Eqs. (1.1) and (1.4) gives the following form for the series

$$
\begin{aligned}
v_y = V_{y,o} &+ V_{y,z}\ r\cos\psi + V_{y,x}\ r\sin\psi \\
&+ V_{y,rr}\left(r^2 - \tfrac{1}{2}R^2\right) + V_{y,rc}\ r^2\cos 2\psi \\
&+ V_{y,rs}\ r^2\sin 2\psi
\end{aligned}
\tag{1.6}
$$

where the six relations:

$$
\begin{aligned}
V_{y,o} &= \frac{1}{\pi R^2}\int_o^R\int_o^{2\pi} v_y\ r dr d\psi \\
V_{y,z} &= \frac{4}{\pi R^4}\int_o^R\int_o^{2\pi} v_y(r\cos\psi) r dr d\psi \\
V_{y,x} &= \frac{4}{\pi R^4}\int_o^R\int_o^{2\pi} v_y(r\sin\psi) r dr d\psi \\
V_{y,rr} &= \frac{12}{\pi r^6}\int_o^R\int_o^{2\pi} v_y\left(r^2 - \tfrac{1}{2}R^2\right) r dr d\psi \\
V_{y,rc} &= \frac{6}{\pi R^6}\int_o^R\int_o^{2\pi} v_y(r^2\cos 2\psi) r dr d\psi \\
V_{y,rs} &= \frac{6}{\pi R^6}\int_o^R\int_o^{2\pi} v_y(r^2\sin 2\psi) r dr d\psi
\end{aligned}
\tag{1.7}
$$

Given a three-dimensional correlation model for the velocity component v_y, it is then possible to utilize Eqs. (1.7) to compute the correlation statistics or power spectral densities for the six "indicial" velocity terms: $V_{y,o}$, $V_{y,z}$, etc. Before proceeding to do this, however, we will first consider the convergence properties of the series.

In general, the convergence of a series based on orthogonal functions requires that the true function be square integrable over the domain of interest (7). The turbulent velocity component, v_y, is a random variable depending on space and time, so that the usual Riemann integration does not apply. The theorems of stochastic integration (8) can be used instead, and the concept of convergence of the series can be defined so that the variance of the difference between the true value and that given by the truncated series goes to zero as more and more terms in the series are included (9). Since the variance of this approximation error is positive over the whole domain, a necessary and sufficient condition for convergence of the series is that the error variance, averaged over the domain, goes to zero. This averaged error variance is then a measure of the convergence properties of the series. Table 1.1 shows the relative approximation error for the truncated series defined by Eq. (1.6).

$$\varepsilon_2 = \frac{1}{\sigma^2 A} \int E[(v_y - \hat{v}_y)^2] \, dA \tag{1.8}$$

where $\hat{v}_y$ = truncated series representation of v_y

σ^2 = variance of v_y

A = area of rotor disk.

The relative approximation error is seen to depend on the dimensionless parameter R/L where R is the disk radius and L is the turbulence integral scale. The computation was carried out using the three-dimensional Von Karman correlation function for isotropic turbulence (10).

Also shown in Table 1.1 are the relative approximation errors when only the uniform term $V_{y,o}$ is retained and when the uniform and shear terms $V_{y,o}$, $V_{y,z}$, and $V_{y,x}$ are retained. These relative approximation errors are designated ε_0 and ε_1, respectively. It can immediately be seen from the table that the quadratic terms improve the approximation and that the approximation is relatively poor when the disk radius approaches the turbulence integral scale. It must be remembered, however, that the Von Karman model does not account for the effects of high wave number viscous dissipation and that the aerodynamic wind turbine rotor forces are always given by spatial integrations which also provide low-pass wave number filtering. Thus, it is expected that these aerodynamic forces will be computed more accurately using the truncated series approximation than is indicated by the data in Table 1.1.

Using uniform and linear gradient terms to approximate the in-plane velocity components yields six turbulence input terms which vary with time. The complete turbulence model can then be written in the following form:

Normal Velocity Components:

$$v_y(x,-V_wt,z) = V_{y,o} + V_{y,z}(z) + V_{y,x}(x) + V_{y,rr}\left(z^2 + x^2 - \tfrac{1}{2}R^2\right) + V_{y,rc}\left(z^2 - x^2\right) + V_{y,rs}(2zx) \tag{1.9}$$

In-Plane Velocity Components:

$$\begin{aligned} v_z(x,-V_wt,z) &= V_{z,o} + \gamma_{zx}x + \bar{\gamma}_{zx}x + \varepsilon_{zx}z + \bar{\varepsilon}_{zx}z \\ v_x(x,-V_2t,z) &= V_{x,o} - \gamma_{zx}z + \bar{\gamma}_{zx}z - \varepsilon_{zx}x + \bar{\varepsilon}_{zx}x \end{aligned} \tag{1.10}$$

where the time-dependent linear gradient turbulence parameters are given by

$$\gamma_{zx} = \frac{1}{2}\left(V_{z,x} - V_{x,z}\right)$$

$$\bar{\gamma}_{zx} = \frac{1}{2}\left(V_{z,x} + V_{x,z}\right)$$

$$\varepsilon_{zx} = \frac{1}{2}\left(V_{z,z} - V_{x,x}\right)$$

$$\bar{\varepsilon}_{zx} = \frac{1}{2}\left(V_{z,z} + V_{x,x}\right)$$

There are twelve turbulence inputs which define the turbulence model. These twelve terms are described in Table 1.2. Drawings of typical fluid streamlines are shown in Figure 1.2 for the in-plane gradient terms.

1.4 Filtered Noise Model For Turbulence

Each of these twelve terms are modeled as a stationary exponentially correlated random process, and they are assumed to be uncorrelated with each other; although it can be shown using mass continuity that $V_{y,o}$, $\bar{\varepsilon}_{zx}$ and $V_{y,rr}$ must be correlated. The $\bar{\varepsilon}_{zx}$ and $V_{y,rr}$ terms are relatively small compared with $V_{y,o}$, and are not associated with large aerodynamic forces allowing this simplication without introducing large error. This makes it possible to represent the turbulence inputs in the following way

$$\frac{dx}{dt} = Ax + Bw \qquad (1.11)$$

where x = the vector of system states

w = the vector of independent white noise excitations

A, B = matrices.

The state correlation matrix is defined by

$$R(\tau) = E[x(t + \tau)x^T(t)] \tag{1.12}$$

and is computed from the differential equation (for $\tau > 0$)

$$\frac{d}{d\tau} R = AR \; , \; R(o) = X \tag{1.13}$$

where the covariance matrix X (assuming zero mean) is given by the solution to the Lyapunov equation (11)

$$AX + XA^T + B\,S_w B^T = 0 \tag{1.14}$$

and S_w is the diagonal matrix of noise power spectral densities.

Assuming that the turbulence terms $V_{y,o}$, $V_{y,z}$, etc. form the state of a system in the form of Eq. (1.11), the correlation matrix $R(\tau)$ is given by the various cross correlations among the individual terms. For example,

$$E[V_{y,z}(t+\tau)V_{y,z}(t)] = \left(\frac{1}{\pi R^2}\right)^2 \iint E[v_y(x_1,-V_w(t+\tau),z_1)v_y(x_2,-V_w t,z_2)] \cdot z_1 z_2 \; dA_1 \; dA_2 \tag{1.15}$$

where the integration is over two disks of radius R. The subscripts 1 and 2 refer to coordinates in the two disks, respectively. Given the correlation matrix $R(\tau)$, the matrix A can be computed by integrating Eq. (1.13)

$$R(\infty) - R(o) = A \int_o^\infty R(\tau)\, d\tau \tag{1.16}$$

or

$$A = -X[S_+]^{-1} \tag{1.17}$$

where $S_+ = \int_o^\infty R(\tau)\, d\tau$

$X = R(o)$

and $R(\infty) = 0.$

The B matrix then must satisfy Eq. (1.14) so that

$$B\, S_w\, B^T = -\left(AX + XA^T\right) \tag{1.18}$$

If the noise terms are chosen (for simplicity) to have identical power spectral densities, then

$$BB^T = -\frac{1}{S_w}\left(AX + XA^T\right) \tag{1.19}$$

where S_w is now the scalar PSD of each noise excitation. A unique matrix B can be determined if it is also required to be triangular, the result of which is called the Chloeskii square root matrix (12).

In cases where $R(\tau)$ is diagonal, considerable simplification results. In this case, A and B will both be diagonal and the resulting scalar equations apply:

$$A_k = -\frac{X_k}{S_{+k}}$$

$$B_k = \sqrt{-\frac{2A_k X_k}{S_w}} \tag{1.20}$$

where the subscript indicates the k^{th} diagonal element.

It is convenient to choose the noise power spectral density

$$S_w = \frac{\sigma^2 L}{V_w^3} \tag{1.21}$$

thereby defining the noise vector to be dimensionless. Also, dimensionless parameters can be chosen so that

$$a_* = -\frac{L\ A_k}{V_w} \tag{1.22}$$

$$b_* = \begin{cases} \dfrac{L\ B_k}{V_w^2} & \text{for uniform terms} \\[2ex] \dfrac{RL\ B_k}{V_w^2} & \text{for shear terms} \\[2ex] \dfrac{R^2\ L\ B_k}{V_w^2} & \text{for quadratic terms} \end{cases} \tag{1.23}$$

These parameters only depend on the dimensionless ratio R/L, where again R is the disk radius and L is the turbulence integral scale. The previous work (13) gives a table of values for the a_* and b_* parameters for the uniform and shear values, while the quadratic terms are found in (14). In summary, then, for a given turbine rotor size and turbulence scale, the a_* and b_* parameters are given. Then using the steady wind speed V_w and the turbulent velocity variance σ^2, the dimensional parameters governing the model are then computed.

In order to avoid the inconvenient interpolation necessary in evaluating the model parameters when R/L is not a tabulated value, a regression procedure was utilized to give a formula for calculating the dimensionless parameters. For the uniform terms, the following form was found to describe the data:

$$a_* \text{ or } b_* = k_1 - \frac{k_2 R_*(1 + k_3 R_*)}{(1 + k_4 R_*)} \qquad (1.24)$$

where $R_* = \frac{R}{L}$.

The parameters k_1, etc. were determined as follows:

1. k_1 is given by the limit as $R_* \to 0$, which is either 1, 2 or $\sqrt{2}$.
2. Assuming $k_3 = 0$ and R_* is small, Eq. (1.24) can be rearranged so that

$$a_* \text{ or } b_* = k_1 - k_2 R_* + k_2 k_4 R_*^2 \qquad (1.25)$$

   the parameters k_2 and k_4 can be found using standard linear regression using the data for small R_*.
3. The equation is then rearranged into the form

$$a_* \text{ or } b_* = k_1 + c_1 \frac{k_4 R_*}{1 + k_4 R_*} + c_2 R \qquad (1.26)$$

   and the parameters c_1 and c_2 are again determined using standard linear regression with k_1 and k_4 fixed. These values then give the final values of k_2 and k_3 parameters.

Table 1.3 shows the resulting regression parameters for the uniform turbulence terms including the in-plane velocity components described in the previous work (14).

For the shear and quadratic terms a different form was found to fit the data. In this case,

$$a_* \text{ or } b_* = k_1 R_*^{-k_2} + k_3 + k_4 R_* \qquad (1.27)$$

The parameter k_2 was chosen to match the slope of a log-log plot of a_* or b_* vs. R_*. A value of $k_2 = 1$ was found to give good results for a_* and $k_2 = 1/4$ for b_*. The remaining parameters, k_1, k_3 and k_4, were determined by standard linear regression. Table 1.4 gives the resulting values for both the normal and in-plane components for the shear terms and for the normal component quadratic terms. Again the data for the in-plane terms were taken from reference (15). In all cases the maximum deviation of the data from the regression curves was less than 5%.

The model describing the turbulent velocity fluctuations can be summarized in polar coordinates in the following manner

Normal Velocity

$$v_y(r,t,\psi) = V_{y,o} + V_{y,x}(r\sin\psi) + V_{y,z}(r\cos\psi) + V_{y,rr}(r^2 - R^2/2) + V_{y,rc}(r^2\cos2\psi) + V_{y,rs}(r^2\sin2\psi) \tag{1.28}$$

In-Plane Velocities

$$\begin{aligned} v_x(r,t,\psi) &= V_{x,o} + (\bar{\gamma}_{zx} - \gamma_{zx})\, r\cos\psi + (\bar{\varepsilon}_{zx} - \varepsilon_{zx})\, r\sin\psi \\ v_z(r,t,\psi) &= V_{z,o} + (\gamma_{zx} + \bar{\gamma}_{zx})\, r\sin\psi + (\varepsilon_{zx} + \bar{\varepsilon}_{zx})\, r\cos\psi \end{aligned} \tag{1.29}$$

where from Figure 1.1, $z = r\cos\psi$ and $x = r\sin\psi$.

Each of the turbulence terms ($V_{y,o}$, $V_{x,o}, \ldots,$ $V_{y,rs}$) is given by an equation of the form

$$\frac{d}{dt}\left[V_{y,\bullet}\right] + a\, V_{y,\bullet} = bw \tag{1.30}$$

where a and b are defined by

$$a = -\frac{V_w}{L} a_* \tag{1.31}$$

$$b = \begin{cases} (V_w^2/L)\ b_* & \text{for uniform terms} \\ (V_w^2/RL)\ b_* & \text{for shear terms} \\ (V_w^2/R^2L)\ b_* & \text{for quadratic terms} \end{cases} \tag{1.32}$$

where a_* and b_* are given by the regression Eqs. (1.24) or (1.26) and depend on the ratio R/L. The white noise term w for each of the twelve turbulence terms is an independent noise source with PSD = $\frac{\sigma^2 L}{V_w^3}$. A computer program which calculates the values of a and b in Eqs. (1.30) is given as the subroutine ATMOS in Appendix C.

1.5 References

1. Connell, J.R., The Spectrum of Wind Speed Fluctuations Encountered by a Rotating Blade of a Wind Energy Conversion System: Observation and Theory, Battelle Pacific Northwest Laboratory, Report PNL 4083, 1981.

2. Thresher, R.W., et al., Modeling the Response of Wind Turbines to Atmospheric Turbulence, Oregon State University, Dept. of Mechanical Engineering, Report RLO/2227-81/2, Aug. 1981.

3. Kristensen, L. and Frandsen, Sten, "Model for Power Spectra Measured from the Moving Frame of Reference of the Blade of a Wind Turbine," Journal of Wind Engineering and Industrial Aerodynamics, V. 10, 1982, pp. 249-262.

4. Rosenbrock, H.H., Vibration and Stability Problems in Large Turbines Having Hinged Blades, ERA Technology Ltd., Surrey, England, Report C/T 133, 1955.

5. Thresher, R.W. and Holley, W.E., The Response Sensitivity of Wind Turbines to Atmospheric Turbulence, Oregon State University, Dept. of Mechanical Engineering, Report RLO/2227-81/1, May 1981.

6. Abroamowitz, M.A. and Segun, I.A., Handbook of Mathematical Functions, U.S. Nat. Bureau Standards, 1964, p. 790.

7. Isaacson, E. and Keller, H.B., Analysis of Numerical Methods, Wiley, 1966, p. 194.

8. Papoulis, A., Probability Random Variables and Stochastic Processes, McGraw-Hill, 1965, pp. 323-325.

9. Ibid, pp. 260-263.

10. Von Karman, T., "Sur la Theorie Statistique de la Turbulence," Comptes Rendus des Seanes de l'Acadamie de Sciences, v. 226, 1948, pp. 2108-2111.

11. Kwakernaak, H. and Sivan, R., Linear Optimal Control Systems, Wiley, 1972, p. 104.

12. Isaacson (1966) op. cit. p. 54.

13. Thresher, et al., (1981), op cit., pp. 142-145.

14. Ibid, pp. 138-146.

15. Ibid.

16. Ibid.

Table 1.1. Relative Approximation Error for Series Approximation.

R/L	ε_0	ε_1	ε_2
.01	.044	.026	.020
.054	.135	.081	.060
.1	.201	.121	.091
.3	.397	.250	.189
.5	.527	.348	.264
1.0	.724	.527	.411
2.0	.889	.737	.608

Table 1.2. Description of Turbulence Input Terms.

Component	Description
$V_{x,o}$	uniform lateral or side component (in plane)
$V_{y,o}$	uniform longitudinal component along mean wind
$V_{z,o}$	uniform vertical component (in plane)
$V_{y,x}$	lateral gradient of longitudinal velocity
$V_{y,z}$	vertical gradient of longitudinal velocity
γ_{zx}	swirl about mean wind axis (in plane)
$\bar{\gamma}_{zx}$	shear strain rates (in plane)
ε_{zx}	
$\bar{\varepsilon}_{zx}$	dilation (in plane)
$V_{y,rr}$	symmetric quadratic variation
$V_{y,rc}$	quadratic with cos2ψ azmuthial variation
$V_{y,rs}$	quadratic with sin2ψ azmuthial variation

Table 1.3. Regression Parameters for Uniform Turbulence Terms.

		k_1	k_2	k_3	k_4
V_z & V_x	a_*	2.0	2.894	-.1383	2.049
	b_*	2.0	3.290	+.0270	2.054
V_y	a_*	1.0	1.713	-.0790	2.048
	b_*	$\sqrt{2.0}$	2.713	+.01591	2.051

Table 1.4. Regression Parameters for Shear and Quadratic Turbulence Terms.

		k_1	k_2	k_3	k_4
$V_{y,z}$ & $V_{y,x}$	a_*	.3266	1.0	.5953	-.1142
	b_*	.2811	.25	.6450	-.1500
γ_{zx}	a_*	.4343	1.0	.9170	-.1532
	b_*	.2579	.25	.6467	-.1093
$\bar{\gamma}_{zx}$ & ε_{zx}	a_*	.5342	1.0	1.276	-2.147
	b_*	.1167	.25	.7733	-.1284
$\bar{\varepsilon}_{zx}$	a_*	1.654	1.0	1.069	+2.154
	b_*	.3546	.25	.3951	+.2593
$V_{y,rr}$	a_*	1.091	1.0	.0276	+.0686
	b_*	.5508	.25	.6473	-.1365
$V_{y,rc}$ & $V_{y,rs}$	a_*	1.081	1.0	.0279	+.0685
	b_*	.3897	.25	.4567	-.0948

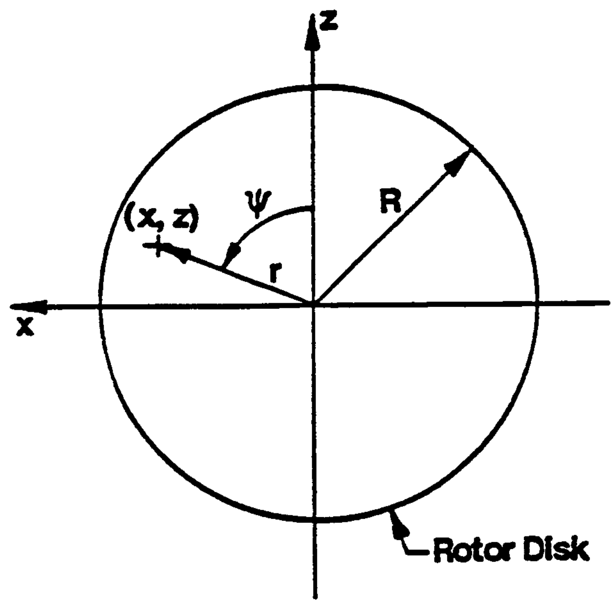

Figure 1.1. Rotor disk coordinate system.

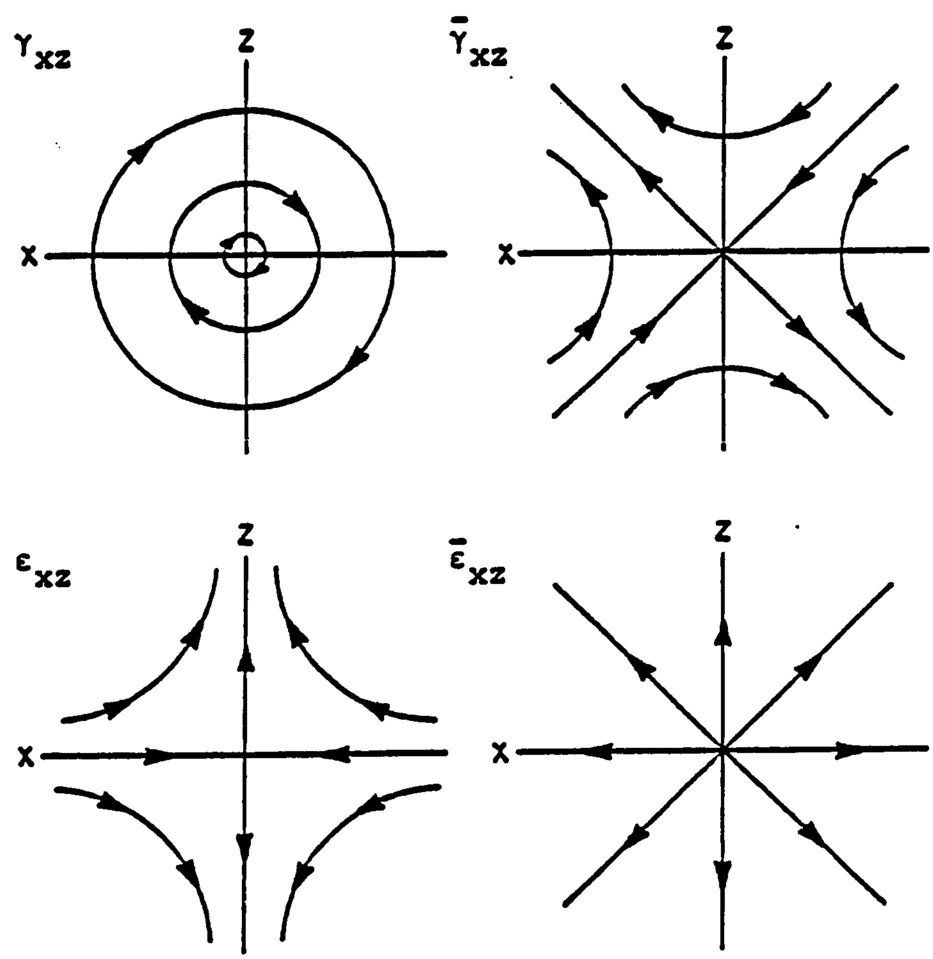

Figure 1.2. Streamlines for in-plane velocity gradient terms.

CHAPTER 2. NUMERICAL SIMULATION

2.1 Introduction

The objective of this chapter is to outline the development for the digital simulation of the turbulence velocity terms. It consists of two parts. First, generation of uniformly distributed random numbers using the multiplicative congruential method to approximate a white-noise time series. Second, generation of the turbulence velocity terms by filtering the white-noise time series to obtain the required shape of the spectral density to produce the appropriate statistics for velocity fluctuations.

2.2 Generation of Uniformly Distributed Random Numbers

There are a number of techniques for generating random variables by digital computers for simulation purposes. Most of these are reproducible and therefore the same sequence of numbers will be generated over and over again given the same starting input. It may be argued that such repeatable random numbers are, in the true statistical sense, deterministic, and not random. Since the digital computer consists of a finite, though large, number of states, the use of an algorithm for the generation of random variables also implies that eventually the computer must return to a state that had existed at the time of some previous implementation of the algorithm which starts the repetition cycle. However, as long as several conditions are met random numbers generated by an algorithm on digital computers can be used for simulation problems. Numbers that are generated by means of a stored algorithm are accordingly referred to as pseudorandom.

Four criteria are usually employed to evaluate the suitability of random number generation method:

1. length of the sequence of the generated random variates,
2. uniformity of amplitude-density spectrum,
3. small degree of autocorrelation, and
4. speed of computer execution.

The first criterion simply means that the period of repetition should be much larger than the intended simulation period. The second implies that a uniform probability density is to be obtained and the degree of the true uniformity is to be a measure of quality. The third condition, if met perfectly, would mean that zero correlation would result, corresponding to true white noise. This is never the case and a reasonably small degree of correlation (and consequent deviation of the power-spectral density from a flat spectrum of white-noise) should be considered allowable.

However, the best criterion is the applicability of the method used to the problem at hand. Methods that are very satisfactory for some applications are found unsuitable when applied to others. With these considerations in mind, the method to be suggested here is the one known either as the multiplicative congruential technique, or as the power residue method. It selects as the k^{th} pseudorandom number the remainder of the division of the product of a constant integer c, and the $(k-1)^{st}$ pseudorandom number by some second constant m. Denoting x_k the k^{th} variate so generated, the operation is described mathematically as follows:

$$x_k = cx_{k-1} \pmod{m} \tag{2.1}$$

where the relation "x (mod m)" denotes the selection of the remainder from the division of x by m. This technique is ideally suited for implementation on a digital computer.

In practice it is recommended that the starting seed value, x_o, be some odd number less than m. For a binary computer, one selects $m = 2^b$ where b is the number of bits per word. The value of the constant c should be of the order m and in the form

$$c = 8k \pm 3 \qquad \text{for any integer } k > 0$$

Thus providing a maximum period of $2^{(b-2)}$ pseudorandom numbers, each between zero and 2^b (1,2). Dividing the generated variates by m gives the numbers between zero and one. This scheme is used in subroutine RANDOM of Appendix C to generate a sequence of uniformly distributed random numbers, starting with an arbitrary selected seed value.

2.3 Construction of a White-Noise Time Series

By definition a set of uniformly distributed random numbers with a range of 0 to m will have a probability density function given by

$$\text{probability density function} \equiv f(x) = \begin{cases} \dfrac{1}{m} & 0 < x < m \\ 0 & \text{otherwise} \end{cases} \qquad (2.2)$$

The mean value and variance of the random variates may be computed from its probability density function, Eq. (2.2), as follows

$$\begin{aligned} \mu_x &= E[X] = \int_{-\infty}^{\infty} x\, f(x)\, dx = \frac{m}{2} \\ \sigma_x^2 &= E[X^2] - [E[X]]^2 = \frac{m^2}{12} \end{aligned} \qquad (2.3)$$

A random time series can be constructed using this set of uniformly distributed random numbers. First, subtract the mean value from each of the variates to obtain a zero mean process, with all values between $-\frac{m}{2}$ and $\frac{m}{2}$. Construct the time series, x(t), by assuming that each of the variates, x_i occurs at intervals Δt apart, and that the value of x(t) is a constant for the period Δt. This produces a random time series x(t), which is a piecewise continuous function of time as illustrated in Figure 2.1. If each number generated, x_i, is statistically independent and therefore uncorrelated with other numbers in the sequence, then the autocorrelation function of x(t) can be determined as

$$R_x(\tau) = E\ [x(t)x(t+2)]$$

$$= \int_0^{\Delta t} x(t)x(t+\tau)f(t)dt$$

$$R_x(\tau) = \sigma_x^2 \left(1 - \frac{|\tau|}{\Delta t}\right) \tag{2.4}$$

This autocorrelation function is plotted in Figure 2.2. Inevitable imperfections in the white-noise properties of the random number generation process are evident by the presence of some degrees of correlation for $|\tau| < \Delta t$.

The corresponding power-spectral density of x(t) may be obtained using the above autocorrelation function as

$$S_x(\omega) \triangleq \int_{-\infty}^{\infty} R_x(\tau)\ e^{-i\omega\tau}\ d\tau$$

$$= \sigma_x^2\ \Delta t\ \left[\frac{1 - \cos\omega\Delta t}{\frac{1}{2}\ (\omega\Delta t)^2}\right] \tag{2.5}$$

which is also plotted in Figure 2.2. If the interval, Δt is sufficiently small (i.e., $\omega\Delta t << 1$), relationship (2.5) becomes approximately

$$S_x(\omega) \approx \frac{m^2\Delta t}{12}\left[1 - \frac{(\omega\Delta t)^2}{12}\right] \tag{2.6}$$

Note that if Δt is selected small enough, with respect to the range of frequencies involved in the simulation problem, it may be considered that the process takes place on the flat part of the spectral curve near $\omega = 0$ (3). For this situation the signal is approximately white-noise with a constant spectral density of

$$S_x(\omega) = \frac{m^2\Delta t}{12} \tag{2.7}$$

2.4 Filter Model

It was shown in Chapter 1 that each of the twelve turbulence terms in the turbulence model can be approximated by an uncorrelated stationary random process. Each term was given by an equation of the form

$$\dot{u} + au = bw \tag{2.8}$$

where
- u = instantaneous value of one of the turbulence terms, $V_{y,o}$, $V_{x,o}$, ..., $V_{y,rs}$
- w = nondimensional zero mean white-noise with power spectral density $S_w = \frac{\sigma^2 L}{V_w^3}$
- σ^2 = turbulent velocity component variance
- L = turbulence integral scale
- V_w = mean wind speed
- R = rotor disk radius.

a and b are given by Eqs. (1.31) and (1.32). The desired power spectral density of the turbulence velocity term is

$$S_u(\omega) = |G(j\omega)|^2 \; S_w = \frac{b^2 S_w}{a^2 + \omega^2} \tag{2.9}$$

where G(s) is the transfer function between the input white-noise, w and output turbulence velocity specified by Eq. (2.8).

To generate a turbulence velocity term digitally let us consider samples of the white-noise forcing function at discrete times t_0, t_1, ..., t_k. Following the procedure outlined in (4), the solution to Eq. (2.8) at time t_{k+1} may be written as

$$u(t_{k+1}) = \phi(t_{k+1}, t_k)\, u(t_k) + \int_{t_k}^{t_{k+1}} b\phi(t_{k+1},\tau)\, w(\tau)\, d\tau$$

and in an abbreviated form

$$u_{k+1} = \phi_k u_k + \bar{w}_k \tag{2.10}$$

ϕ_k is the state transition matrix for the step t_k to t_{k+1}, and $\bar{w}_k$ is the driven response at t_{k+1} due to the presence of the white-noise input during the (t_k, t_{k+1}) interval. Note that the white-noise input required in the continuous model automatically assures that $\bar{w}_k$ will be an uncorrelated white-noise sequence in the discrete model (4).

From Eq. (2.8) the transition matrix is easily determined as

$$\phi_k = e^{-a\Delta t} \tag{2.11}$$

The variance of $\bar{w}_k$ is established by using the convolution integral as

$$\sigma_{\bar{w}}^2 = E[\bar{w}^2] = \int_o^{\Delta t} \int_o^{\Delta t} g(u)g(v)R_f(u-v)\, dudv \tag{2.12}$$

where $g[\cdot]$ = unit impulse response

$$g(t) \triangleq \bar{L}^{1}\ [G(s)] = be^{-at} \tag{2.13}$$

and $R_f[\cdot]$ = autocorrelation function of the input white-noise. The autocorrelation function of the input white-noise can be established as

$$R_f[u-v] = E[w(u)w(v)] = S_w\delta(u-v) \tag{2.14}$$

where S_w is the power spectral density of the input. Substituting Eqs. (2.13) and (2.14) in Eq. (2.12) and carrying out the integration, the variance of $\bar{w}_k$ becomes

$$\sigma_{\bar{w}}^2 = E[\bar{w}^2] = \frac{b^2 S_w}{2a} \left(1 - e^{-2a\Delta t}\right) \tag{2.15}$$

If the generated random signal x(t) with zero mean and variance $\sigma_x^2 = \frac{m^2}{12}$ is used to approximate $\bar{w}(k)$ at the time intervals t_1, t_2, $\ldots t_k$, and if

$$\bar{w}(k) = cx(k) \tag{2.16}$$

then the mean square of both sides is

$$E[\bar{w}^2] = c^2 E[x^2]$$

$$\sigma_{\bar{w}}^2 = c^2 \sigma_x^2$$

Substitute for $\sigma_x^2 = \frac{m^2}{12}$ and solving for c gives

$$c = \left\{\frac{6b^2 S_w}{am^2} \left(1 - e^{-2a\Delta t}\right)\right\}^{1/2} \tag{2.17}$$

Substituting Eq. (2.17) in Eq. (2.16) and using the result and Eq. (2.11) in Eq. (2.10), gives the turbulence velocity term at t_{k+1} as

$$u_{k+1} = e^{-a\Delta t} u_k + \left\{\frac{6b^2 S_w}{am^2} \left(1 - e^{-2a\Delta t}\right)\right\}^{1/2} x_k \tag{2.18}$$

If the range of the random numbers, m, is 1 then Eq. (2.18) can be written as

$$u_{k+1} = e^{-a\Delta t} u_k + \{\frac{6b^2 S_w}{a} (1 - e^{-2a\Delta t})\}^{1/2} x_k \tag{2.19}$$

Evaluating the variance of the generic turbulence term from Eq. (2.9) gives

$$\sigma_u^2 = E[u^2(t)] = R_u(\tau=0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_u(w)dw$$

$$\sigma_u^2 = \frac{b^2 S_w}{2a}$$

Taking the mean square of both sides of Eq. (2.19) gives an identical result.

2.5 References

1. Mirham, G.A., Simulation, Statistical Foundations and Methodology, Academic Press, 1972, Ch. 2.

2. Hull, T.E. and Dobell, A.R., Random Number Generators, SIAM Rev. 4, 1962, pp. 230-254.

3. Kochenburgber, R.J., Computer Simulation of Dynamic Systems, Prentice-Hall, 1972, pp. 405-445.

4. Brown, R.G., Random Signal Analysis and Kalman Filtering, John Wiley, 1983, pp. 181-211.

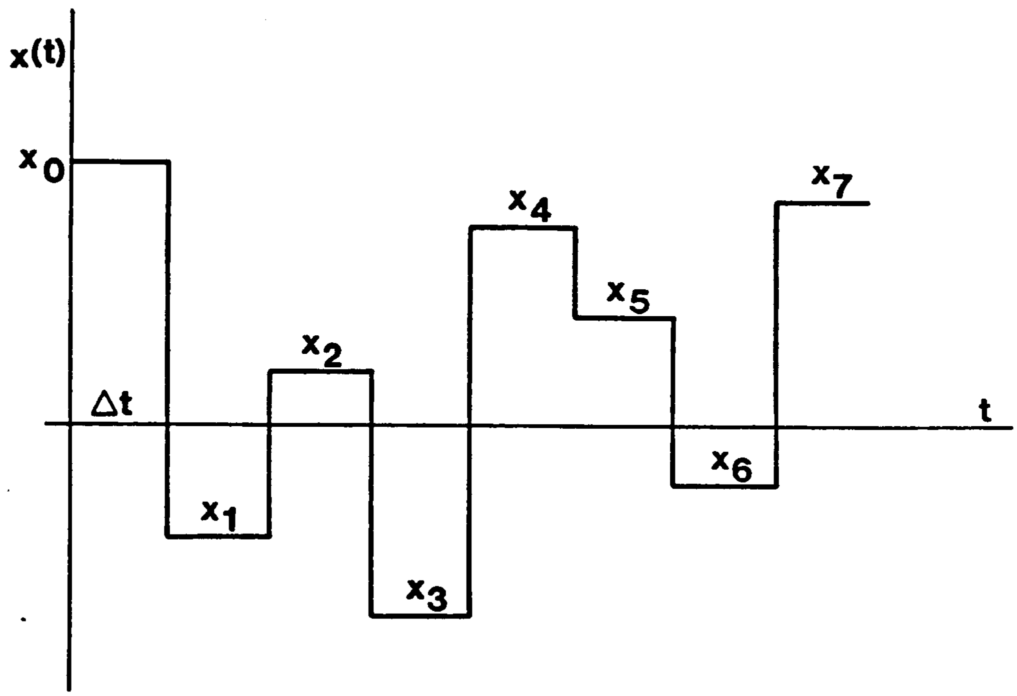

Figure 2.1. Time series constructed from a sequence of uniformly distributed random numbers.

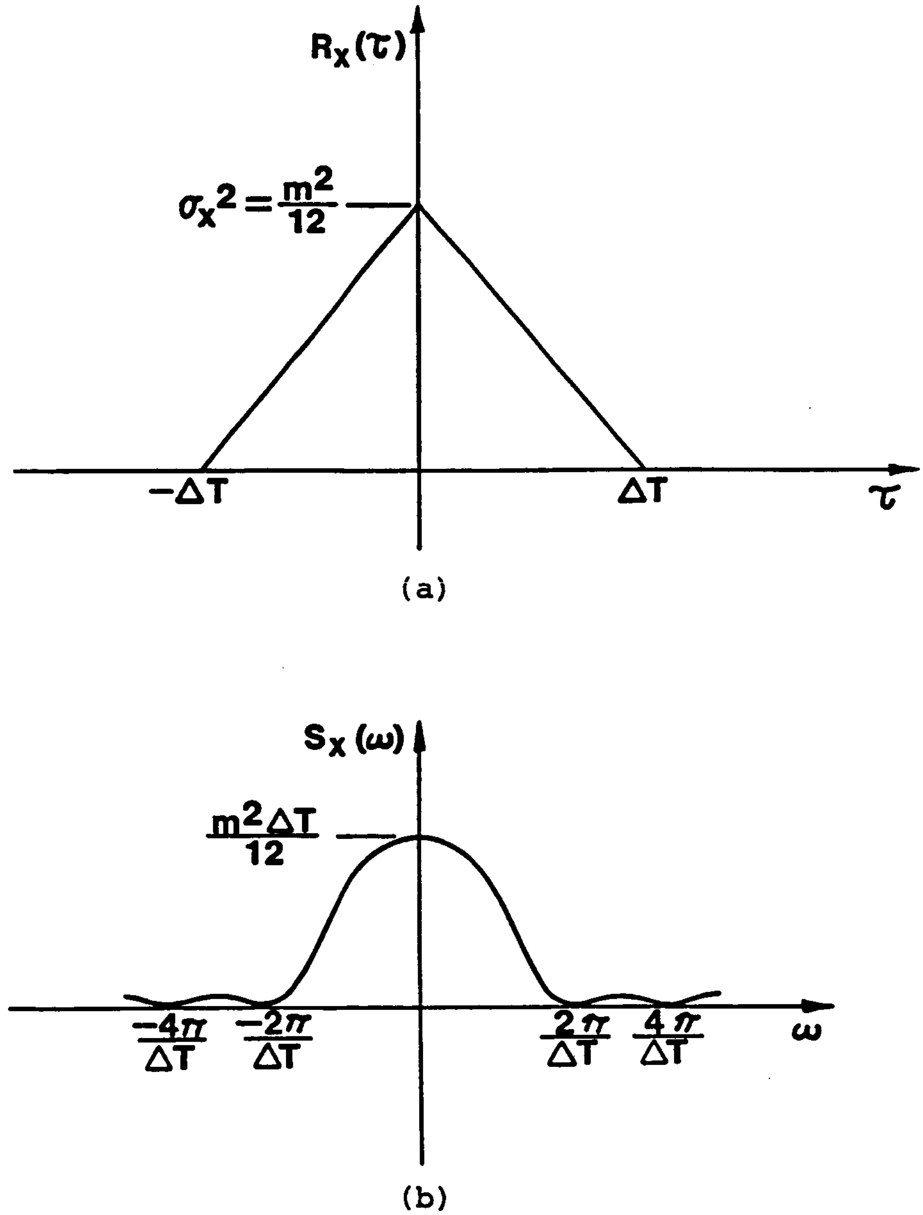

Figure 2.2. (a) Autocorrelation function and,

(b) Spectral density function for the constructed time series.

CHAPTER 3. DIGITAL COMPUTER IMPLEMENTATION

3.1 Introduction

In this chapter the computer code for digital simulation of turbulence velocity components is discussed. The program is written in Fortran V on the CDC Cyber 170/720 series. It is designed in a block-structured form so the various tasks performed within the program are essentially separate routines and are linked together by an executive main program. It is run interactively but can be run in a batch mode with some prior preparation of response data.

3.2 Input Data

A list of the input variables is given in Table 3.1. The user has the opportunity to change any of the input variables listed in Table 3.1 at execution time. When a run is completed the program allows the user to either end execution with the current data set, recycle the current data file with different values for the input variables, or employ a new data file.

3.3 Computer Algorithm for Turbulence Simulation

It was shown in Chapter 1, Eqs. (1.28) and (1.29), that turbulence velocity components can be given in polar coordinates in the following form:

Normal Velocity

$$v_y\ (r,\psi,t) = V_{y,o} + V_{y,x}\ (r\sin\psi) + V_{y,z}\ (r\cos\psi)$$

$$+ V_{y,rr}\left(r^2 - \frac{R^2}{2}\right) + V_{y,ro}\ (r^2\cos 2\psi)$$

$$+ V_{yors}\ (r^2\sin 2\psi) \tag{1.28}$$

In-Plane Velocities

$$v_x\ (r,\psi,t) = V_{x,o} + \left(\bar{\gamma}_{zx} - \gamma_{zx}\right) r\cos\psi + \left(\bar{\varepsilon}_{zx} - \varepsilon_{zx}\right) r\sin\psi$$

$$v_z\ (r,\psi,t) = V_{z,o} + \left(\bar{\gamma}_{zx} + \gamma_{zx}\right) r\sin\psi + \left(\bar{\varepsilon}_{zx} + \varepsilon_{zx}\right) r\cos\psi \tag{1.29}$$

where each turbulence term ($V_{y,o}$, $V_{x,o}$, ..., $V_{y,rs}$) is given by an equation of the form

$$\frac{d}{dt}\left[V_{y,\cdot}\right] + aV_{y,\cdot} = bw$$

with a and b given by Eqs. (1.31) or (1.32).

The simulation routine SIMULX generates the appropriate a and b coefficients based on the given input data and the curve fitting contained in subroutine ATMOS. The procedure for obtaining these coefficients is described in Section 1, and the regression method is described in Appendix A. Next, the subroutine TURBS actually simulates the velocity fluctuations by first calling RANDOM to generate a white noise time signal as discussed in

Section 2.3. This signal is then filtered using Eq. (2.16) to obtain the twelve turbulence parameters of Table 1.2, $V_{x,o}$, $V_{y,o}$, $V_{z,o}$, $V_{y,k}$, etc. The values of these twelve turbulence parameters are then substituted into Eqs. (1.28) and (1.29) to obtain the resulting velocity fluctuations, v_x, v_y, and v_z, at any desired radial station for the current time. As the procedure marches forward in time, the blade moves to a new azimuth angle and subroutine TURBS is called again to repeat the procedure. A flow chart of this process is shown in Figure 3.1 for the executive program SIMULX, and Figure 3.2 shows the flow chart for subroutine TURBS.

The number of points along the blade at which turbulence velocity is evaluated is given as the parameter, NPTS, in the program SIMULX, and can be easily changed. The turbulence velocity components then are computed at equally spaced points along the blade from an initial radius to a final radius which the user specifies. For the results presented here, only one radial position at the tip was considered (NPTS = 1). As much as possible, the code has been written to contain its own documentation through extensive use of comments within the program.

Appendices C through F include a complete program listing, a sample input data file, a procedural example of the interactive features, and the results of the sample run as observed from the tip of a Mod-0A wind turbine blade.

3.4 Tool Kit for Signal Analysis

Analysis of random signals requires some basic mathematical tools. There are two general methods of describing random signals mathematically. The first, and more basic, is a probabilistic description in which the random quantity is characterized by a probability model. However, it tells very little about how the random signal varies with time, or how the amplitude varies as a function of frequency.

For this work dealing with atmospheric turbulence it is helpful to use some of the typical statistical measures to charactrize the wind signal using the mean, variance, correlation function, and spectral density. These measures allow the signal which is being simulated to be compared with various theoretical models and with experimental data. This is essential because when comparing wind turbine responses generated using a simulated wind with responses obtained from field test measurements the comparison must be made for the "same" atmospheric conditions. This means that the mean, variance, and spectral density for the simulated wind should match those of the real atmosphere during the field test period. The tools for computing these statistical parameters are discussed in this section.

Subroutine MEANVAR estimates mean and variance of a time series. Since each of the turbulence velocity components is computed by low pass filtering of a uniformly distributed white noise time series, it is expected that the resulting turbulent velocity fluctuations will have nearly a Gaussian distribution

(1). To estimate the actual distribution subroutine ROB constructs a frequency histogram which can be compared with the standard normal distribution.

Subroutine PSD generates spectral density estimates of the generated velocity signals. It uses a fast Fourier transform (FFT) algorithm to calculate discrete Fourier transforms (DFT) (2). A cosine tapered data window is used to smooth the data at each end of the record before it is analyzed (which has the effect of sharpening the spectral window). In order to improve the accuracy of the results, the signal is broken into a number of segments and the spectral estimates for each segment are computed and then averaged for all segments at each frequency. A more detailed discussion of the digital signal analysis is given in Appendix B.

In order to obtain accurate estimates of the spectral density, relatively long sequences of random velocities are needed. The length of each of the time series segments in the code is set by the parameter LSPECT, which has been arbitrarily set equal to 128 in a parameter statement. It can easily be changed but must always equal an integer power of 2 for the FFT algorithm to work properly. The user specifies the number of random velocities generated as the input parameter, NRVELOC. The user can choose any size up to 6500, the dimension size of the array. Note that if NRVELOC is not evenly divisible by the segment length, LSPECT, then an appropriate number of zeros will be added to each time series. This might make the length of the time series exceed the declared array size for the velocity components. To avoid this, NRVELOC should be kept smaller than the velocity time series array size minus LSPCT, (currently NRVELOC $<$ 6500 -128). Because of larger array sizes it might not be feasible to run this program interactively on some computers. Therefore, modification may be required depending on the needs and resources available to the user.

3.5 References

1. Newland, D.E., An Introduction to Random Vibration and Spectral Analysis, Longman, 1975, pp. 80-81.
2. Ibid, Appendix 2.

Table 3.1. List of Input Variables.

CONST	constant coefficient in the power residue algorithm (subroutine RANDOM) for generation of uniformly distributed random numbers
DELTAT	time step interval for generation of random velocity components (sec)
DIVIDER	module used in function (mod) (•) in the power residue algorithm (subroutine RANDOM)
SEED	initial random number used in the power residue algorithm (subroutine RANDOM)
NRVELOC	number of elements of random turbulence velocity component sequences
OMEGA	rotor speed (rpm)
OMEGAZ	initial angular orientation in the rotor disk plane (deg)
ROTR	rotor radius (feet)
RRATIO	ratio of radial position to blade radius
TI	turbulence intensity ($\frac{\sigma}{V_w}$ in percent)
TL	turbulence integral scale (feet)
VRANGE	number of standard deviations displayed for the turbulent velocity probability density function (usually selected to be 3)
V_w	mean wind velocity (mph)

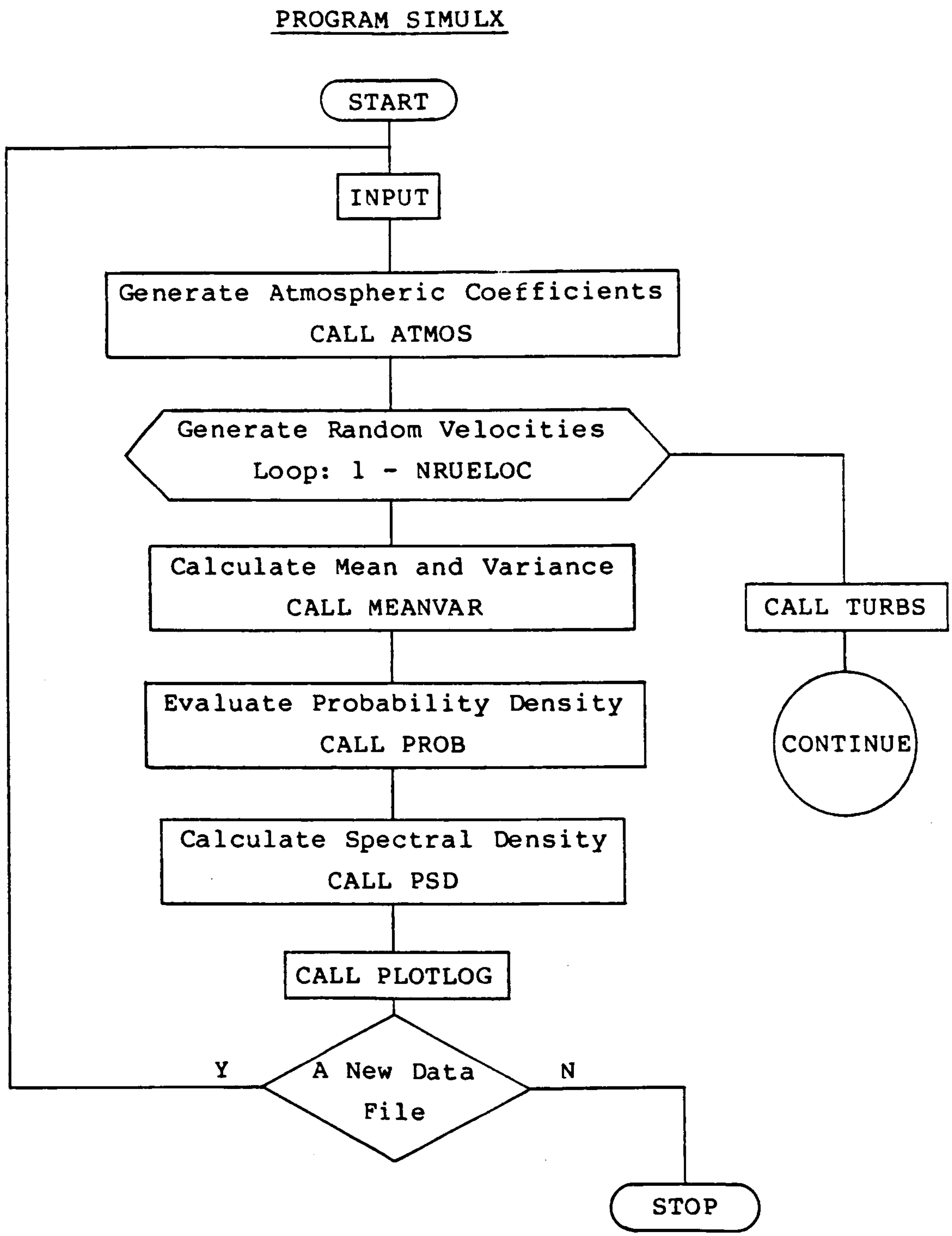

Figure 3.1. Flow chart of the program SIMULX.

SUBROUTINE TURBS

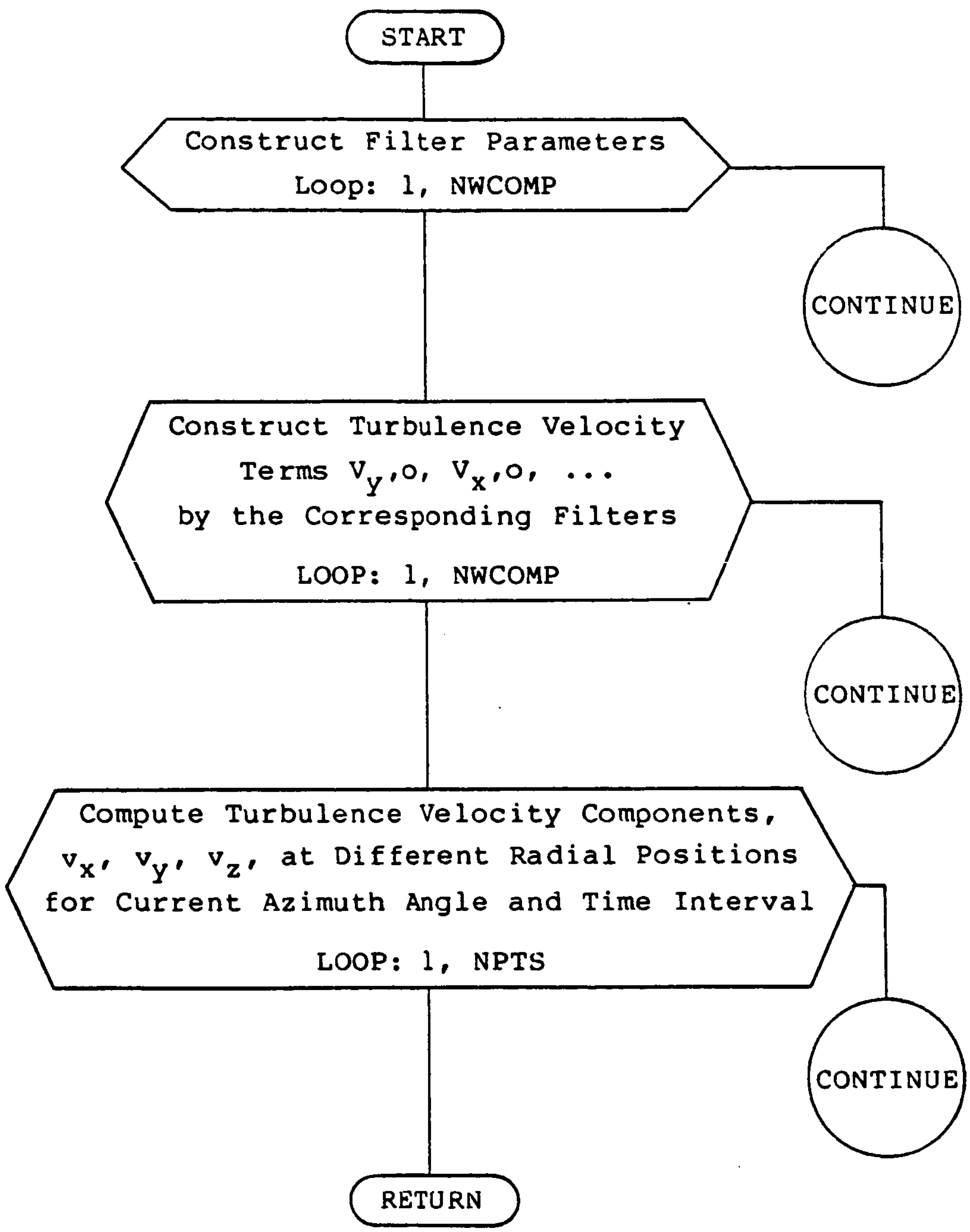

Figure 3.2. Flow chart of subroutine TURBS

CHAPTER 4. SIMULATION RESULTS

4.1 Introduction

This chapter presents some typical results obtained using the computer code to simulate the turbulence inputs for wind turbines. Simulation results are presented for two wind turbine sizes. The first turbulence simulation is for the Mod-0A, 200 kW wind turbine, which has a rotor diameter of 125 ft. The spectral density of the simulated turbulence is compared with field test data taken from the vertical plane array experiments of George and Connell (1), for similar wind conditions. In addition, the results are compared with the theoretical Von Karman spectra for the atmospheric boundary layer. The second simulation is for a Mod-2, 300-ft diameter wind turbine. In this case, there is no appropriate test data which can be used for comparison, but a comparison is made with the Von Karman spectrum for the longitudinal velocity component.

4.2 Comparison of Simulations

Figure 4.1 shows the simulation time series of the longitudinal velocity component, V_y, as observed from the tip of a rotating Mod-0A blade. In this simulation, the tip radius was taken as 62.5 ft and the rotor speed was 40 rpm. In addition, the parameters used for the turbulence simulation where V_w = 26.25 ft/s, σ/V_w = 0.10 and the turbulence integral scale, L, was 400 ft. In Figure 4.1, the mean wind speed has been removed. Figure 4.2 presents the spectral density for the time series shown in Figure 4.1. The simulated spectrum clearly shows the spikes at 1 and 2 cycles per rotor revolution that are the result of rotation of the blade through the wind turbulence field. However, the simulation results show no spikes higher than 2 cycles per revolution because the model only allowed for velocity fluctuation harmonics up to $\sin 2\Psi$ and $\cos 2\Psi$ as indicated by Eq. (1.28). The data taken from the vertical plane array is plotted showing harmonics up to 3 cycles per rotor revolution, but higher harmonics are present in the original presentation by George and Connell (1). The simulation results show considerably greater spectral energy in the frequency range of .1 to .3 hz than the VPA results. This is probably because the a* an b* coefficients used to generate the simulation were selected so that the Von Karman spectrum would be approximated in the low frequency range. As is shown in the figure, the comparison with the Von Karman spectrum in this frequency range is quite good. It would be possible to more closely approximate the vertical plane array data by adjusting the a* and b* coefficients for the $V_{y,o}$ term of Eq. (1.28). In addition, it would be possible to add additional harmonics to the model in order to obtain the 3 and 4 cycles per revolution spectral spikes, but that would involve a significant effort. It is hoped that some experience with the implementation of the existing model in a dynamics code could be obtained, before attempting to improve the simulation, and account for these additional effects.

Figure 4.3 shows the probability density function for the time series of the V_y turbulent velocity fluctuations. As can be seen from the figure, the simulated velocity fluctuations closely approximate a Gaussian distribution.

Figure 4.4 is a spectral density plot for the vertical velocity component, V_z, as provided by the simulation. The Von Karman spectrum for this turbulence component is also provided for comparison. The simulation is for the case where the turbulence is observed from the tip of a rotating Mod-0A blade. Whereas the Von Karman spectrum plotted is for a point fixed in space. The simulated spectrum shows a single spike at a frequency of 1 cycle per rotor revolution. Theoretically there should be many of these spikes each at a multiple of the rotor blade passage frequency. However, the simplified simulation model, Eq. (1.29), for the in-plane velocity components includes only the first harmonic. No field data is available for comparison of the in-plane velocity components. The simulation spectrum for the lateral velocity component was virtually identical to the results for the vertical component and therefore has not been presented. Figure 4.5 shows the probability density function for the time series of the V_z velocity fluctuations, and the figure shows the distribution to be approximately Gaussian.

Figure 4.6 is the spectral density plot of the longtidinual velocity component, V_y, for a simulation run for a Mod-2 sized turbine. In this simulation, the mean wind speed was V_w = 32.15 ft/s, σ/V_w = .061 and the turbulence integral scale was taken as 500 ft. The velocity field was simulated at two radial locations along the rotor blade. One was at 30% span and the other was for the 70% span location. This illustrates one of the convenient features of this turbulence model. At each time step, the velocity fluctuations at all radial locations are obtained simultaneously, as can be seen by the form of Eq. (1.28). Figure 4.6 includes the Von Karman spectrum for comparison. Figure 4.7 shows a probability density plot for the velocity fluctuations at 70% span. Unfortunately, there is no appropriate test data with which to compare these simulation results a the Mod-2 sized turbine.

4.3 Concluding Remarks

The authors offer the following conclusions and remarks on the basis of the work presented in this report:

1. The results presented here show that the turbulence simulation model does a reasonable job of representing many of the features of atmospheric turbulence.
2. The turbulence simulation model presented here does not model the spectral spikes in the wind input above 2 cycles per rotor revolution. If these spectral spikes at higher harmonics turn out to be important for cyclic load prediction then this model will be incomplete. It should be noted that this model does contain some spectral energy

at the higher harmonics of rotor speed; it is the effect of rotating through the turbulence structure that is missing at the higher frequencies.

3. The great advantage of this model is its simple structure and fast computation speed. This simulation model will not significantly increase the complexity of a wind turbine dynamic model.

The authors hope that in the near future, this model will be used to generate inputs for a structural dynamic model, so that its usefulness in predicting cyclic loads can be assessed. Ability to predict cyclic loads reasonably well for a small computational cost is the ultimate goal, and this simulation approach seems to offer promise of achieving that goal.

4.4 Reference

1. George, R.L. and Connell, J.R., Rotationally Sampled Wind Characteristics and Correlations with MOD-0A Wind Turbine Response, Pacific Northwest Laboratory Report PNL-5238, September 1984.

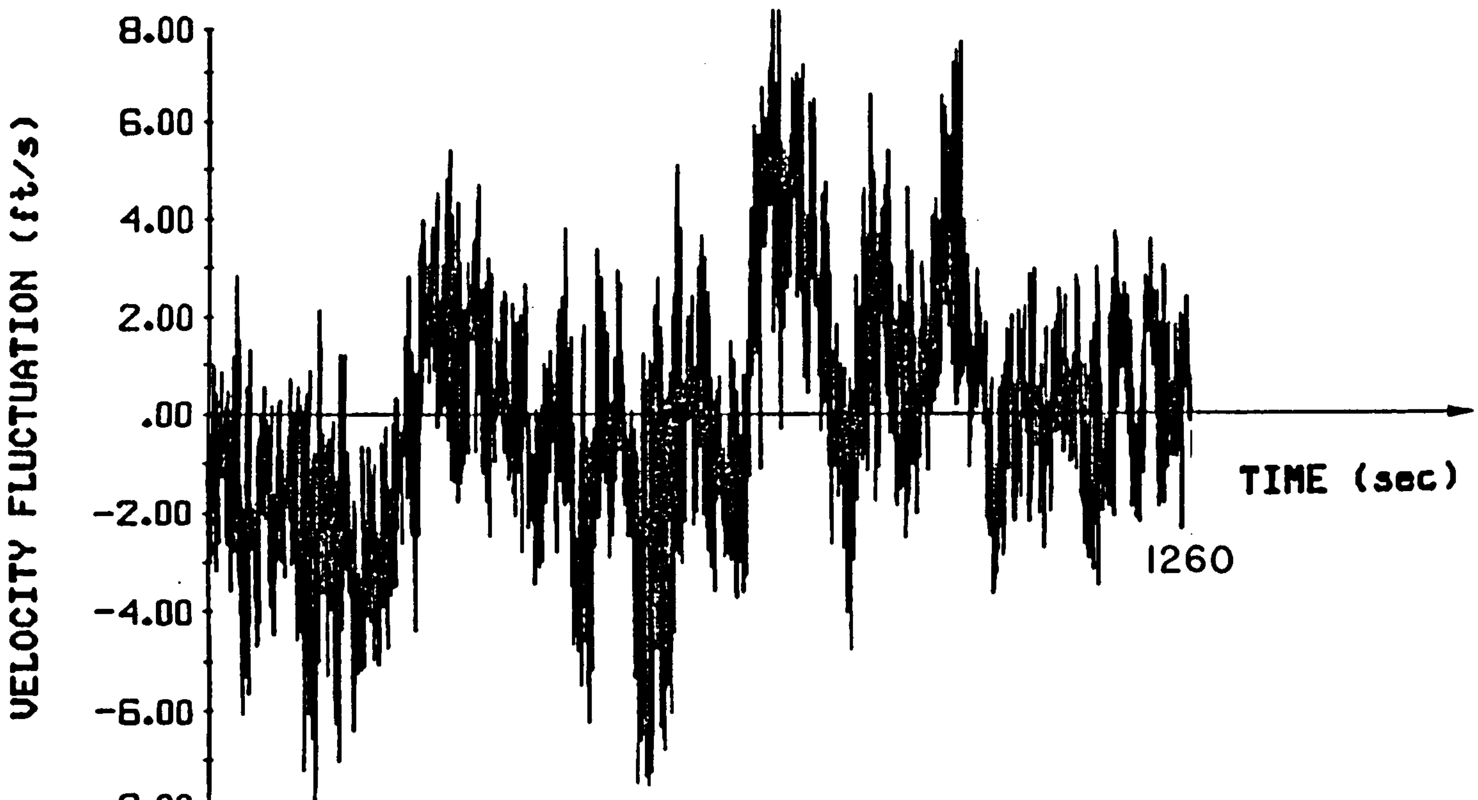

Figure 4.1. Simulated time series for the longitudinal, velocity component, V_y, as observed from the tip of a rotating Mod-0A blade with V_w = 26.25 ft/s, σ/V_w = .10 and L = 400 ft.

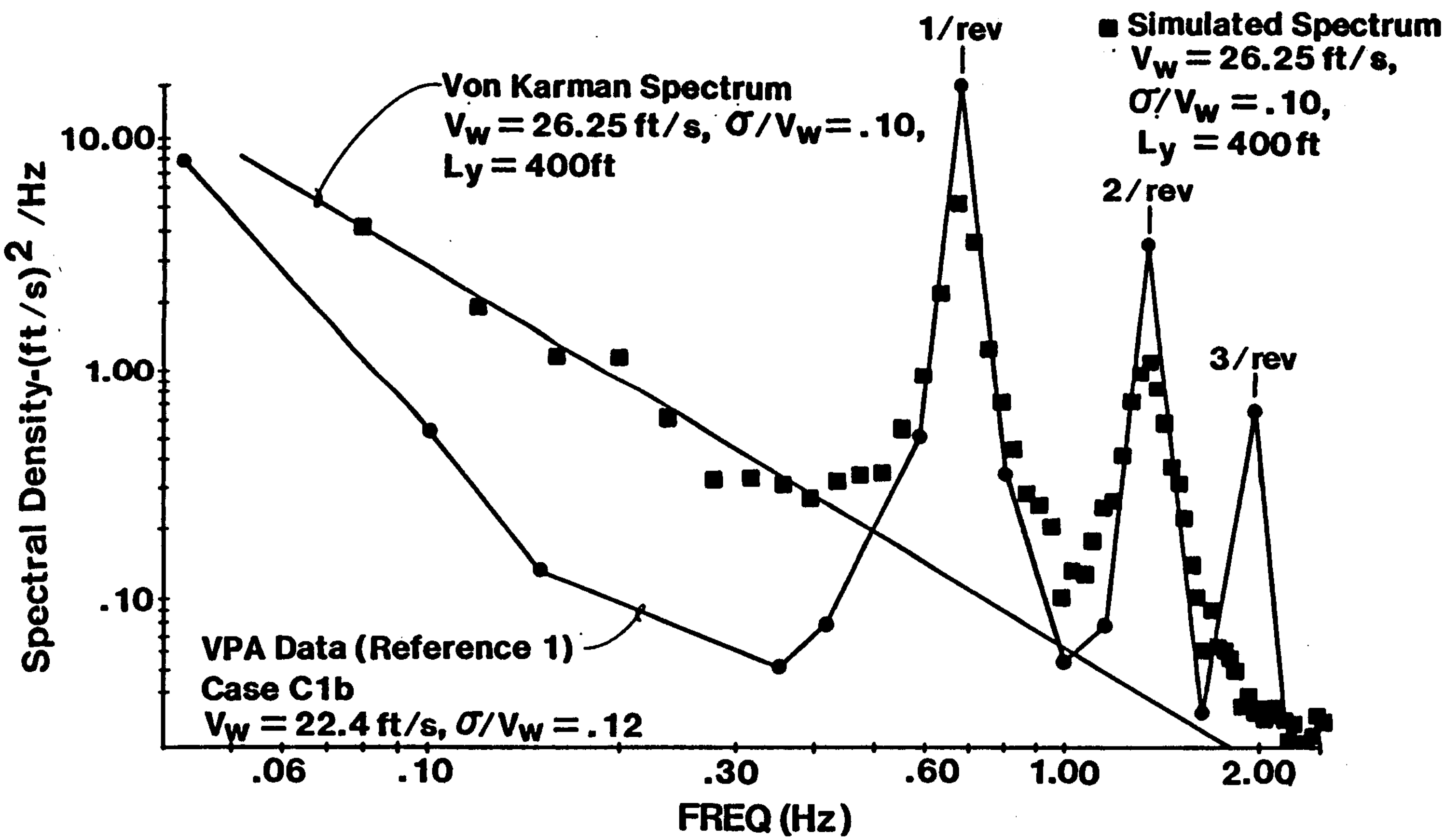

Figure 4.2. Spectral density plot of the V_y simulated time series as observed from the tip of a Mod-0A wind turbine blade.

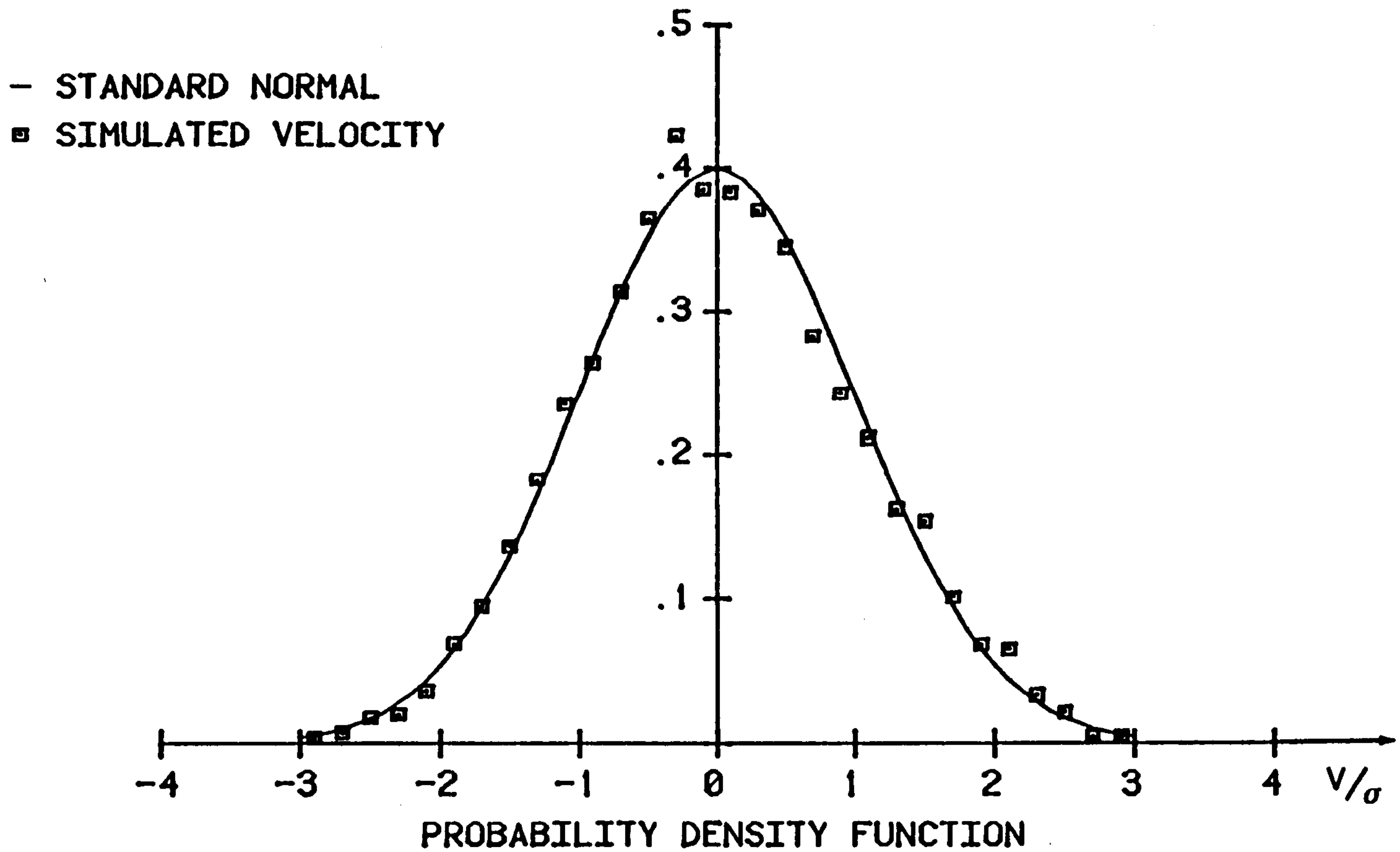

Figure 4.3. Probability density for the Mod-0A turbine simulation, V_y component.

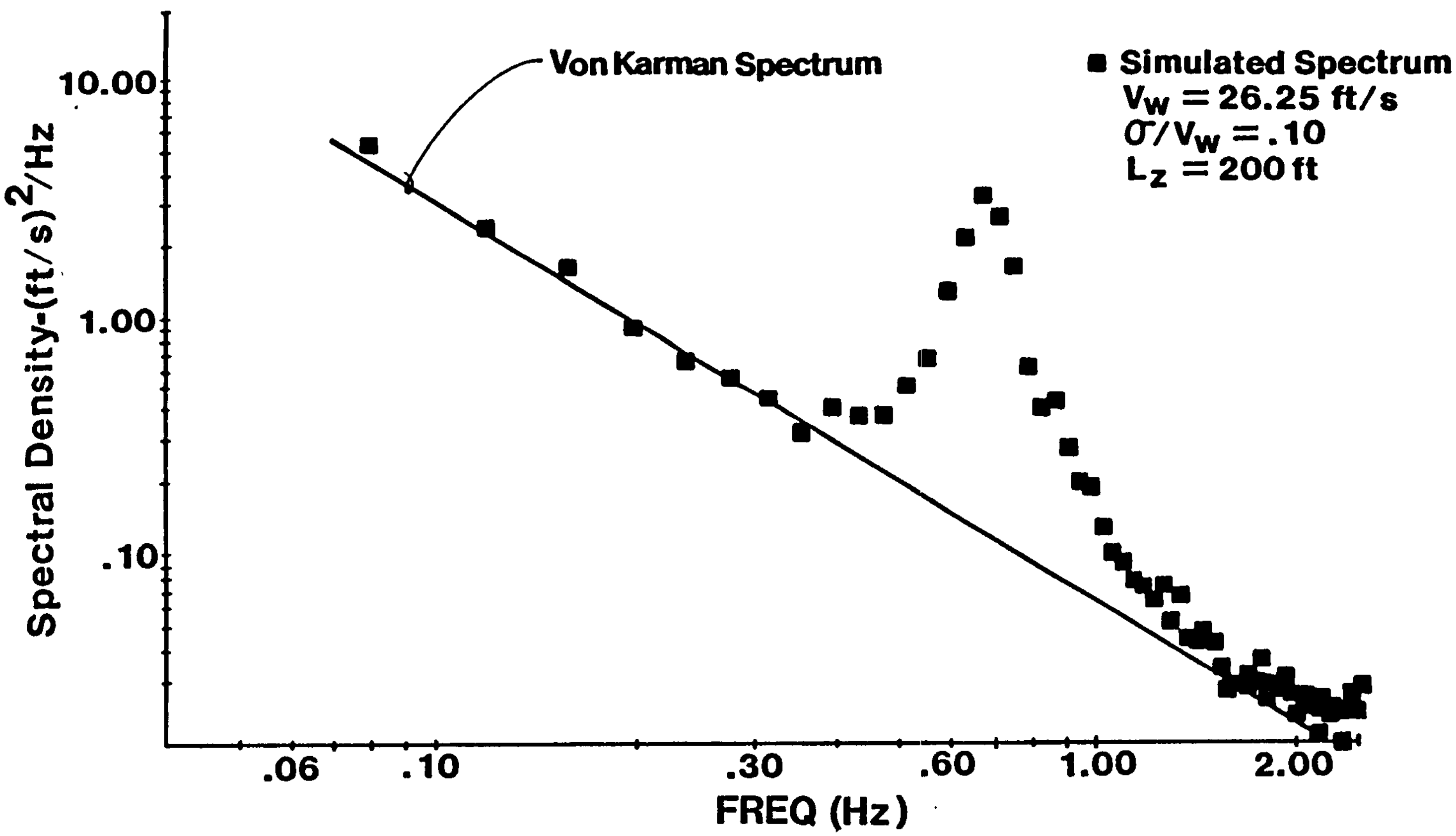

Figure 4.4. Spectral density plot of the V_z simulated time series as observed from the tip of a Mod-0A wind turbine blade.

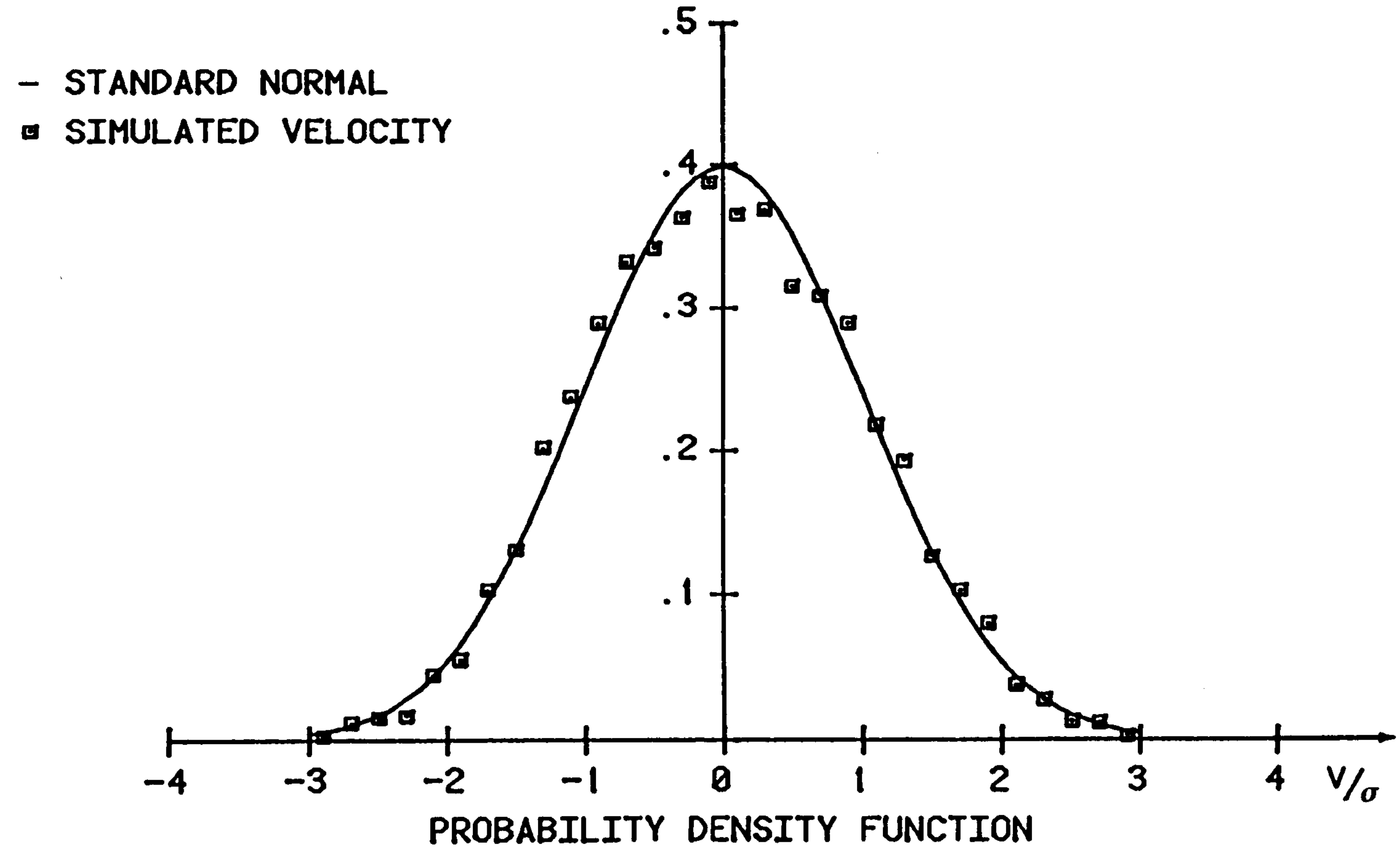

Figure 4.5. Probability density for the Mod-0A turbulence simulation, V_z component.

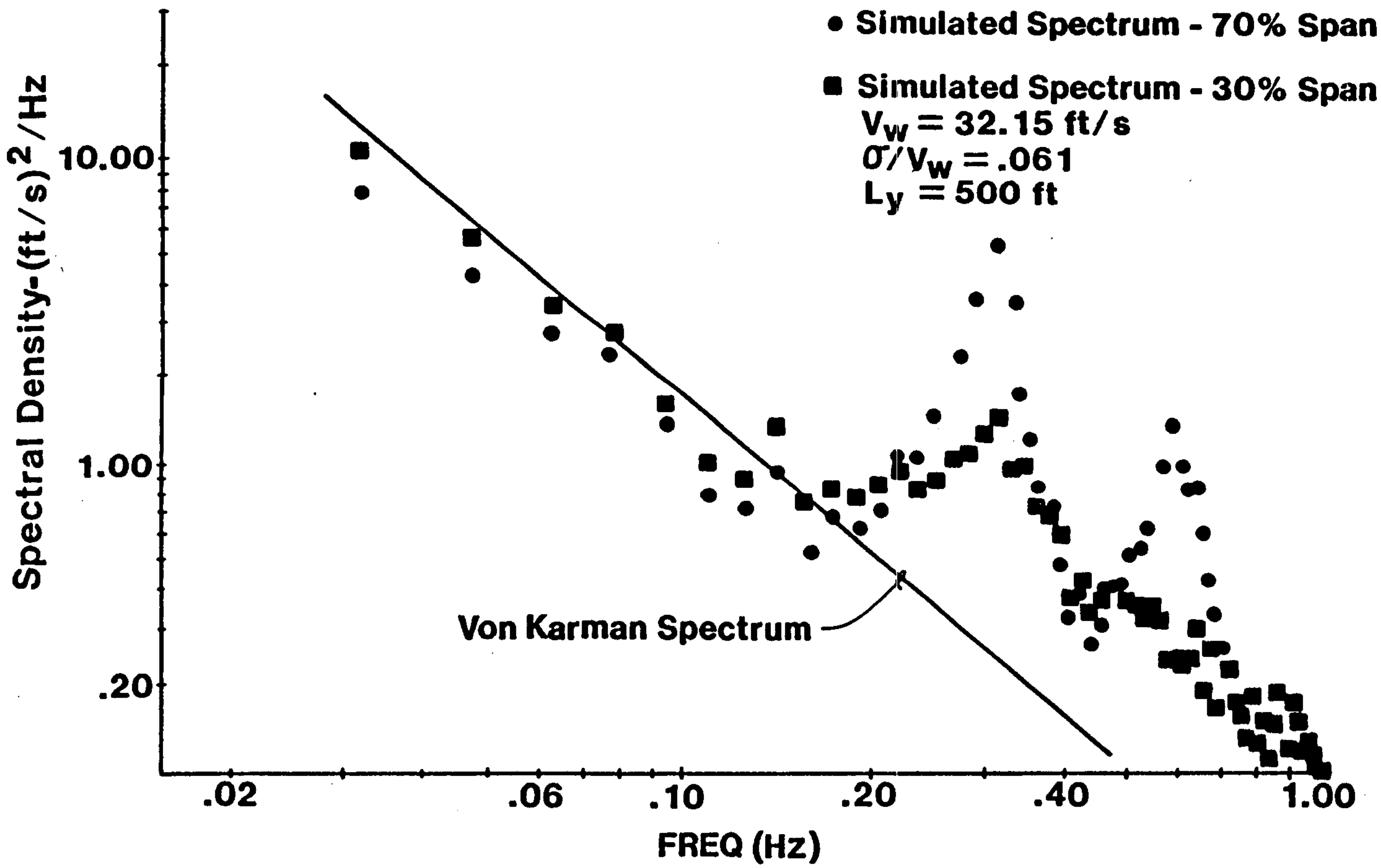

Figure 4.6. Spectral density plot of the V_y turbulence component as observed from a Mod-2 rotor blade at different radial stations.

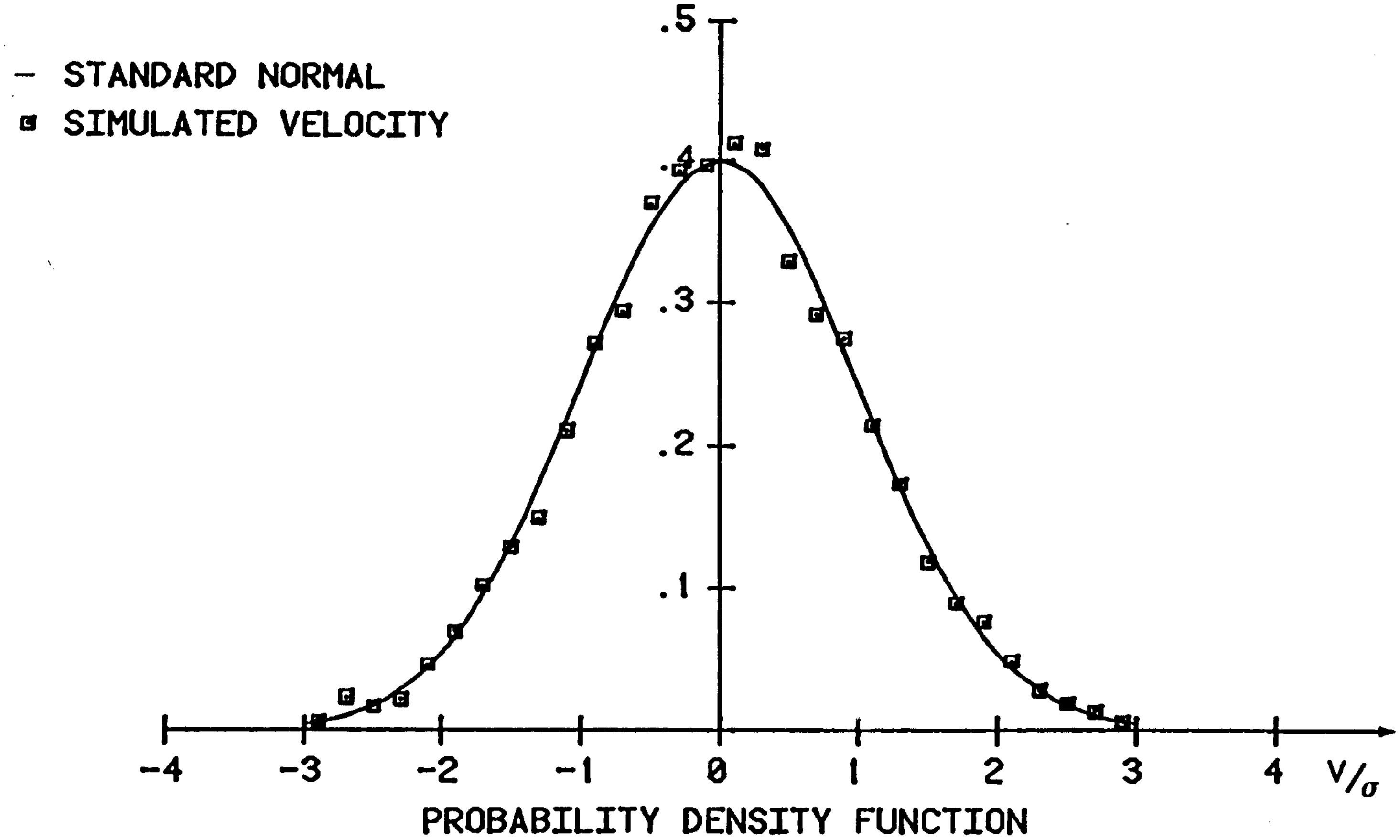

Figure 4.7. Probability density for the V_y component of the Mod-2 simulation at 70% span.

APPENDIX A. LINEAR LEAST-SQUARES REGRESSION (1)

For the general regression problem, the form of the relation

$$y = f(x,a) \tag{A.1}$$

where: x = independent variable

a = vector of parameters

y = dependent variable

is known and it is desired to determine the vector of parameters, a, when several data points (x_i, y_i) are given. In the case when the parameters appear linearly, i.e.,

$$y = a_1 f_1(x) + a_2 f_2(x) + \ldots + a_n f_n(x) \tag{A.2}$$

the data parameters form a set of linear equations given by

$$\sum_{j=1}^{n} f_j(x_i) a_j = y_i \qquad i = 1,\ldots,m \tag{A.3}$$

When there are more data points than unknown parameters (i.e., $m > n$) the equations are overdetermined and it is unlikely that all equations can be satisfied exactly. When $m < n$ the equations are underdetermined and many different sets of parameter values can be found which fit the data exactly. To determine a reasonable solution to the problem, the parameters can be chosen to minimize the sum of the squares of the residuals, i.e.,

$$\mathrm{Min}_a \sum_{i=1}^{m} (y_i - f(x_i,a))^2 \tag{A.4}$$

It can be shown (2), in the case when the data are given exactly by

$$y_i = F(x,a_*) + e_i \tag{A.5}$$

where a_* are the true parameters and e_i are mutually independent random errors which are normally distributed with zero mean, that the least-squares solution is equivalent to choosing the most probable values of a, given the data (assuming no prior knowledge of a). In cases when there are more parameters than data (i.e., $m < n$) it is reasonable to set the last n-m parameters to zero then to determine the remaining m parameters which fit the data exactly.

In order to find the least squares solution, it is convenient to put the problem in matrix form

$$y - Fa = e \tag{A.6}$$

where

$$F = \begin{matrix} f_1(x_1) & f_2(x_1) & \cdots \\ f_1(x_2) & \cdot & \\ \vdots & & \ddots \end{matrix}$$

$$y = \begin{matrix} y_1 \\ \vdots \\ y_m \end{matrix}$$

e = residual vector (dimension m)

The necessary conditions for the minimum are easily found by differentiating to be

$$2\left(\frac{\partial e}{\partial a}\right)^T e = 0 \tag{A.7}$$

or using Eq. (A.6) and the definition of F

$$2(-F)^T(y-Fa) = 0$$

or, finally

$$(F^TF)a = F^Ty \tag{A.8}$$

The solution is unique when the matrix F^TF is nonsingular.

Instead of solving Eq. (A.8) directly for

$$a = (F^TF)^{-1}F^Ty \tag{A.9}$$

Golub (3) suggested using the Householder (4) decomposition of the matrix F, i.e.,

$$F = QR \tag{A.10}$$

where Q is orthogonal and R has all elements below the diagonal equal to zero. Thus, Eq. (A.8) can be rewritten as

$$(QR)^T(QR)a = (QR)^Ty \tag{A.11}$$

or since Q is orthogonal (i.e., $Q^{-1} = Q^T$)

$$(R^TR)a = R^TQ^Ty \tag{A.12}$$

for the case when m > n, R is of the form

$$R = \begin{matrix} U \\ \cdots \\ O \end{matrix}$$

where U is upper triangular, and the coefficient matrix for a becomes

$$R^T R = U^T U \tag{A.13}$$

Now, let the right hand side be partitioned so that

$$Q^T y = \begin{matrix} z_1 \\ \cdots \\ z_2 \end{matrix} \tag{A.14}$$

Since U and F have the same rank = n, Eq. (A.14) becomes

$$Ua = z_1 \tag{A.15}$$

The solution to Eq. (A.15) involves only a simple back substitution since U is triangular.

This procedure has been implemented in a standard library subroutine supplied by the IMSL (5) and is utilized to compute the regression parameters in the turbulence model.

A.1 References

1. Lawson, C.L. and Hanson, R.J., Solving Least Squares Problems, Prentice-Hall, 1974.
2. Goodwin, G.C. and Payne, R.L., Dynamic System Identification, Academic Press, 1977, pp. 29-41.
3. Golub, G., "Numerical Method for Solving Linear Least Squares Problems," Numerische Mathematik, V. 7, 1965, pp. 206-216.
4. Housholder, A.S., "Unitary Triangularization of a Nonsymmetric Matrix," J. Assn. Comput. Mach., V. 5, 1958, pp. 339-342.
5. Anon., Documentation for Routine LLSQF, The International Mathematical and Statistical Library, IMSL, Inc., Houston, TX.

APPENDIX B. DIGITAL SPECTRAL ANALYSIS

The power spectral density of a stationary random process x(t) is defined as

$$S_x(\omega) = \int_{-\infty}^{\infty} R_x(\tau)\, e^{-i\omega t}\, dt \tag{B.1}$$

where $R_{x(\tau)}$ is the autocorrelation function of x(t) given by

$$R_x(\tau) = E\,[x(t)x(t+\tau)] \tag{B.2}$$

If the random process x(t) is sampled at intervals Δ (constant) then the discrete value of x(t) at time $t = r\Delta$ is written x_r and the sequence $\{x_r\}$, r = 0, 1, 2, ..., is called a discrete time series. The objective of time series analysis is to determine the statistical characteristics of the original function x(t) by manipulating the discrete time series $\{x_r\}$. The main interest is the frequency composition of x(t). For this, the power spectral density of x(t) is estimated by analyzing the discrete time series obtained by sampling a finite segment of x(t). Discrete Fourier transform (DFT) of a time series $\{x_r\}$, r = 0, 1, 2, ..., (N-1) is defined as follows:

$$X_k = \frac{1}{N} \sum_{r=0}^{N-1} x_r\, e^{-i\left(\frac{2\pi k}{N}\right) r} \qquad k = 0, 1, 2, \ldots, (N-1) \tag{B.3}$$

and the inverse discrete Fourier transform (IDFT) is given by

$$x_r = \sum_{k=0}^{N-1} X_k\, e^{i\left(\frac{2\pi r}{N}\right) k} \qquad r = 0, 1, 2, \ldots, (N-1) \tag{B.4}$$

where the range of the Fourier components X_k is limited to $k = 0$ to $(N-1)$ corresponding to harmonics of frequency $\omega_k = \frac{2\pi k}{T} = \frac{2\pi k}{N\Delta}$ where $T = N\Delta$ is the finite segment of the sampling function $x(t)$ and Δ is the sampling interval.

It can be shown (1) that the spectrum of $x(t)$ can be estimated by $\tilde{S}(\omega_k)$ as follows

$$\tilde{S}(\omega_k) \approx TS_k \tag{B.5}$$

where S_k is the DFT of the discrete autocorrelation R_r which for two random processes $x(t)$ and $y(t)$ and their corresponding sampled time series $\{x_r\}$ and $\{y_r\}$ is given by

$$R_r = \frac{1}{N} \sum_{S=0}^{N-1} x_s y_{s+r} \qquad r = 0, 1, 2, \ldots, (N-1) \tag{B.6}$$

Substituting for x_r and y_r from (B.4) it is possible to demonstrate that S_k can be obtained as

$$\begin{aligned} S_{xxk} &= X_k^* \, X_k \\ S_{xyk} &= X_k^* \, Y_k \\ S_{yxk} &= Y_k^* \, X_k \\ S_{yyk} &= Y_k^* \, Y_k \end{aligned} \tag{B.7}$$

where the complex conjugate of X and Y are denoted at X* and Y*.

The fast Fourier transform subroutine listed in Reference (1) is used to evaluate the DFT's of the time series. The FFT works by partitioning the full sequence $\{x_r\}$ into a number of shorter sequences. Instead of calculating DFT of the original sequence, only the DFT's of the shorter sequences are computed and then averaged to yield the full DFT of $\{x_r\}$. A cosine data taper function is used to smooth the data at each end of the data record before carrying out the DFT to improve the shape of the resulting spectral density (2,3).

B.1 References

1. Newland, D.E., An Introduction to Random Vibrations and Spectral Analysis, Longman (1975).
2. Cooley, J.W., Lewis, P.A., and Welch, P.D. The Application of the Fast Fourier Transform Algorithm to the Estimation of Spectra and Cross Spectra. IBM Research Paper, IBM Watson Research Center, Yorktown Heights, NY, 1967.
3. Otnes, R.K. and Enochson, L., Digital Time Series Analysis, John Wiley, New York, 1972.

APPENDIX C. COMPUTER CODE LISTING

Listing of the program SIMULX.

```
      PROGRAM SIMULX (INPUT,OUTPUT)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C                                                                    C
C  PROGRAM SIMULX GENERATES THE WIND TURBULANCE AT POINTS ALONG THE  C
C  BLADE IN THE ROTOR DISK AND FINDS THE FREQUENCY SPECTRUM OF EACH  C
C  VELOCITY COMPONENT.  A UNIFORRMLY DISTRIBUTED RANDOM NUMBER IS    C
C  GENERATED TO SIMULATE WHITE NOISE.  EACH TURBULENCE VELOCITY TERM C
C  MODELED AS A STATIONARY RANDOM PROCESS GIVEN BY AN EQUATION OF    C
C  THE FORM                                                          C
C                                                                    C
C           D(U) / DT + A * U = B * W                                C
C                                                                    C
C  WHERE  W  : NON-DIMENSIONAL ZERO MEAN WHITE NOISE WITH POWER      C
C                SPECTRAL DENSITY SW.                                C
C           A; : ATMOSPHERIC PARAMETER CONSTANTS.                    C
C           B                                                        C
C  SOLUTION TO THIS EQUATION FOR A DISCRETE TIME WHITE NOISE CAN BE  C
C  WRITTEN AS                                                        C
C                                                                    C
C           U(K+1) = PHI(K,K+1) * U(K) + W(K)                        C
C                                                                    C
C  WHERE    U(K);         : SOLUTIONS AT TIMES T(K); T(K+1)          C
C           U(K+1)                                                   C
C           PHI(K),K+1)  : TRANSITION FUNCTION FROM TIME T(K)        C
C                           TO T(K+1)                                C
C           W(K)          : DRIVEN RESPONSE AT T(K+1) DUE TO THE     C
C                           PRESENCE OF WHITE NOISE INPUT DURING TIME C
C                           T(K), T(K+1) INTERVAL.  NOTE THAT W(K) IS C
C                           A WHITE NOISE RANDOM SEQUENCE.           C
C  SUBROUTINE ATMOS GENERATES THE ATMOSPHERIC CONSTATNTS PARAMETERS  C
C  A'S AND B'S.  SUBROTINE RANDOM GENERATES A SEQUENCE OF UNIFORMLY  C
C  DISTRIBUTED RANDOM NUMBERS WHILE ROUTINE MEANVAR CALCULATES MEAN  C
C  AND VARIANCE OF TIME SERIES.                                      C
C  SUBROUTINE PSD IS USED TO GENERATE THE SPECTRUM OF THE GENERATED  C
C  SIGNALS. STANDARD PLOT OF RANDOM VELOCITY VS TIME IS OBTAINED     C
C  USING SUBROUTINE PLTSTND.  SUBROUTINE PLTLOG PROVIDES LOG-LOG     C
C  PLOT FOR SPECRUM VS FREQUENCY.                                    C
C                                                                    C
C  NOTE: IF THE NUMBER OF GENERATED RANDOM VELOCITY                  C
C        COMPONENTS, NRVELOC, IS NOT EVENLY DIVISIBLE BY             C
C        LENGTH OF THE SPECTRUM, LSPECT, THEN NRVELOC                C
C        MUST BE SMALLER THAN THE DECLARED SIZE OF RANDOM            C
C        VELOCITY COMPONENT ARRAYS AT MOST BY LSPECT SO              C
C        AFTER PADDING THE TIME SERIES IT IS NOT OVER SIZED.         C
C                                                                    C
C                                                                    C
C                                                                    C
```

```
C                                                                    C
C  LIST OF ARGUEMENTS:                                               C
C  CONST   : CONSTATNT COEFFICIENT IN THE POWER RESIDUE ALGORITHM    C
C            (SUBROUTINE RANDOM) FOR GENERATION OF UNIFORMLY         C
C            DISTRIBUTED RANDOM NUMBERS                              C
C  DIVIDER : MODULE USED IN FUNCTION MOD(.) IN SUBROUTINE RANDOM     C
C  SEED    : INITIAL RANDOM NUMBER USED IN THE POWER RESIDUE         C
C            ALGORITHM, SUBROUTINE RANDOM                            C
C  NWCOMP  : NUMBER OF TURBULENT VELOCITY TERMS IN THE ATMOSPHERIC   C
C            MODEL                                                   C
C  NRVELOC : NUMBER OF ELEMENTS OF RANDOM TURBULENT VELOCITY         C
C            COMPONENTS SEQUENCE                                     C
C  NPTS    : NUMBER OF POINTS ALONG THE BLADE AT WHICH TURBULENT     C
C            VELOCITY IS EVALUATED                                   C
C  NBINS   : NUMBER OF SUBINTERVALS ON THE POSITIVE VELOCITY         C
C            AXIS FOR DETERMINING PROBABILITY DISTRIBUTION           C
C  PROBDIS : ARRAY OF SIZE (2*NBINS) WHICH CONTAINS PROBABILITY      C
C            DISTRIBUTION OF THE TURBULENT VELOCITY COMPONENTS       C
C            IN EACH SUBINTERVAL(BIN)                                C
C  VRANGE  : MAXIMUM VALUE OF TURBULENT VELOCITY AS AN INTEGER       C
C            MULTIPLE OF ITS VARIANCE, SUBROUTINE PROB               C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INTEGER NWCOMP,NPTS,LSPECT,LP2,NRVELOC
      INTEGER NBINS,NLABEL,CONST
      PARAMETER (LSPECT=128,LP2=7)
      PARAMETER (NWCOMP=12,NPTS=1,NLABEL=1,NBINS=16)
      REAL R,ROTR,OMEGA,OMEGAZ,DELTAT,DIVIDER,VRANGE
      REAL VX(6500),VY(6500),VZ(6500),Y(6500),X(200)
      REAL PROBDIS(2*NBINS)
      REAL XX(NPTS),YY(NPTS),ZZ(NPTS)
      REAL A(NWCOMP),B(NWCOMP),CC(NWCOMP),DD(NWCOMP)
      REAL PSY(LSPECT/2+1),F(LSPECT/2+1),SOUT(LSPECT/2+1)
      COMPLEX ZY(LSPECT)
      DOUBLE PRECISION SEED
      CHARACTER *7 FILEIN, FILEOUT, LABEL(NLABEL)*40
      CHARACTER *2 ANS1, ANS*1
      COMMON /TURBINE/ OMEGA,OMEGAZ,ROTR
      COMMON /WIND/ TL,TI,SW,VW
      COMMON /ATMOS/ A,B
      COMMON /RAND/ CONST, SEED, DIVIDER
      NAMELIST /INDATA/ CONST,DELTAT,DIVIDER,SEED,NRVELOC,OMEGA,
     &                  OMEGAZ,ROTR,RRATIO,TI,TL,VRANGE,VW
*
* .... CONVERSION FACTORS ....
*
      PI      = ACOS(-1.)
      CDEGRAD = PI/180.
      CRPMRPS = 2.*PI/60.
```

```
      CMPHFPS = 5280./3600.
*
* .... INTERACTIVE : SELECT INPUT AND OUTPUT FILES,
*                    OPEN FILES, READ DATA FILE. USE NAMELIST.
 881  PRINT *, ' '
      PRINT *, 'ENTER NAME OF THE NEW DATA FILE '
      READ '(A)', FILEIN
      OPEN (5,FILE=FILEIN)
      PRINT *, ' '
      PRINT *, 'ENTER THE NAME OF OUTPUT FILE '
      READ '(A)', FILEOUT
      OPEN (6,FILE=FILEOUT)
*
* .... INPUT ....
*
* .... READ THE PLOT LABELS ....
*
      DO 100 I=1,NLABEL
         READ (5,'(A)') LABEL(I)
 100  CONTINUE
      READ (5,INDATA)
      REWIND (5)
      CLOSE (5)
*
* .... PRINT ECHO OF INPUT DATA ....
*
 1    PRINT *, ' '
      PRINT 5, 'CONST   =', CONST
      PRINT 6, 'DELTAT  =', DELTAT  ,  '(SEC)     '
      PRINT 7, 'DIVIDER =', DIVIDER
      PRINT 7, 'SEED    =', SEED
      PRINT 5, 'NRVELOC =', NRVELOC
      PRINT 6, 'OMEGA   =', OMEGA   ,  '(RPM)     '
      PRINT 6, 'OMEGAZ  =', OMEGAZ  ,  '(DEG)     '
      PRINT 6, 'ROTR    =', ROTR    ,  '(FEET)    '
      PRINT 6, 'RRATIO  =', RRATIO
      PRINT 6, 'TI      =', TI      ,  '(PERCENT) '
      PRINT 6, 'TL      =', TL      ,  '(FEET)    '
      PRINT 6, 'VRANGE  =', VRANGE
      PRINT 6, 'VW      =', VW      ,  '(MILES/HR)'
 5    FORMAT (1X,A15,I8)
 6    FORMAT (1X,A15,F12.3,T35,A15)
 7    FORMAT (1X,A15,E20.13)
*
* .... INTERACTIVE: CHANGE DATA VALUES & REPEAT ECHO CHECK OR CONTINUE ..
*
      PRINT *, ' '
      PRINT *, 'DO YOU WANT TO CHANGE ANY VALUES ? ENTER(Y OR N)'
      READ '(A)', ANS
      IF (ANS .EQ. 'Y') THEN
         PRINT *, ' '
```

```
            PRINT *, 'TO CHANGE VALUES, LEAVE COLUMN 1 BLANK AND TYPE'
            PRINT *, '$INDATA FOLLOWED BY VALUE ASSIGNMENTS IN THE FORM:'
            PRINT *, 'NAME = VALUE, NAME = VALUE ,..., $'
            PRINT *, 'NOTE : COLUMN 1 MUST BE BLANK; TERMINATE WITH $ '
            READ INDATA
            PRINT *, ' '
            GO TO 1
         ENDIF
*
* ..... UNIT CONVERSIONS : (RPM) TO (RAD/SEC); (DEG) TO (RAD) ....
*                          (MPH) TO (FT/SEC)
*
      OMEGA  = OMEGA  * CRPMRPS
      OMEGAZ = OMEGAZ * CDEGRAD
      VW     = VW     * CMPHFPS
*
      WRITE (6,10) CONST,SEED,DIVIDER
 10   FORMAT(//,5X,'POWER RESIDUE METHOD WITH THE FOLLOWING PARAMETERS'
     &       ,/,5X,'IS USED TO GENERATE UNIFORMLY DISTRIBUTED RANDOM '
     &       ,'NUMBERS',//,10X,'CONSTANT COEFF, CONST',T35,'= ',I8,/,10X
     &       ,'SEED',T35,'= ',1X,E15.8,/,10X
     &       ,'MODULE DIVIDER, DIVIDER',T35,'= ',1X,E20.13)
*
* ....
* .... GENERATE ATMOSPHERIC COEFFICIENTS ....
* ....
      CALL ATMOS
* ....
* .... PRINT ATMOSPHERIC COEFFS ....
      WRITE (6,15)
 15   FORMAT(//,20X,'ACOEFF',12X,'BCOEFF')
       DO 140 I=1,NWCOMP
         WRITE (6,20) I,A(I),B(I)
 20      FORMAT(/,5X,I5,5X,E13.6,5X,E13.6)
 140  CONTINUE
* ....
* .... GENERATE RANDOM VELOCITIES ...
* ....
      R       = RRATIO * ROTR
      ANGSTEP = DELTAT * OMEGA
      IF (NPTS .EQ. 1) THEN
         BEGINR = R
         FINR   = R
      ELSE
         NSEG=NPTS-1
         PRINT *, 'NO. OF SEGEMENTS ALONG THE BLADE, NSEG= ',NSEG
         PRINT *, 'NO. OF POINTS ALONG THE BLADE WHERE VELOCITY '
         PRINT *, 'COMPONENTS ARE CALCULATED, NPTS= ',NPTS
         PRINT *, 'ENTER THE BEGINNING AND FINAL RADIUS ALONG THE '
         PRINT *, 'BLADE, BEGINR, AND FINR.'
         READ  *, BEGINR,FINR
```

```
      ENDIF
      DO 200 J=1,NRVELOC
         PSI=J*ANGSTEP+OMEGAZ
* ....
         CALL TURBS (XX,YY,ZZ,DELTAT,BEGINR,FINR,NPTS,PSI)
* ....
         VX(J)= XX(1)
         VY(J)= YY(1)
         VZ(J)= ZZ(1)
 200  CONTINUE
* .... CALCULATE MEAN AND VARIANCE OF THE TIME SERIES
* ....
      CALL MEANVAR (VXMEAN,VXVAR,VX,NRVELOC)
      CALL MEANVAR (VYMEAN,VYVAR,VY,NRVELOC)
      CALL MEANVAR (VZMEAN,VZVAR,VZ,NRVELOC)
* ....
* ....
      WRITE (6,25) NRVELOC,DELTAT,R,OMEGA,OMEGAZ,VW,TL,TI,SW
     &             ,VXMEAN,VXVAR,VYMEAN,VYVAR,VZMEAN,VZVAR
 25   FORMAT(//,10X,'NUMBER OF RANDOM VELOCITIES GENERATED, NRVELOC'
     &        ,T65,'= ',I5,/,10X,'TIME STEP TO GENERATE THE RANDOM '
     &        ,'VELOCITY, DELTAT',T65,'= ',E12.5,/,10X,'RADIAL DISTANCE '
     &        ,'TO SELECTED POINT ALONG THE ROTOR, R',T65,'= ',E12.5,/
     &        ,10X,'ROTOR SPEED, OMGA',T44,'= ',E12.5,T64,'(RAD/SEC)',/
     &        ,10X,'INITIAL ROTATION, OMEGA-ZERO',T44,'= ',E12.5,T64
     &        ,'(RAD)',/,10X,'WIND VELOCITY, VW',T44,'= ',E12.5,T64
     &        ,'(FEET/SEC)',/,10X,'TURBULENCE INTEGRAL SCALE, TL',T44
     &        ,'= ',E12.5,T64,'(FEET)',/,10X,'TURBULENCE INTENSITY, '
     &     ,'TI',T44,'= ',E12.5,T64,'(PERCENT)',/,10X,'SPECTRUM OF THE '
     &     ,'INPUT WHITE NOISE, SW =',T65,'= ',E12.5,5X,'(SEC)',/,10X
     &     ,'MEAN VALUE OF VX =',E14.7,4X,'VARIANCE OF VX =',E14.7,/,10X
     &     ,'MEAN VALUE OF VY =',E14.7,4X,'VARIANCE OF VY =',E14.7,/,10X
     &     ,'MEAN VALUE OF VZ =',E14.7,4X,'VARIANCE OF VZ =',E14.7)

      DO 220 J=1,NRVELOC
         VX(J)=VX(J)-VXMEAN
         VY(J)=VY(J)-VYMEAN
         VZ(J)=VZ(J)-VZMEAN
 220  CONTINUE
* ....
      PRINT *, 'TO GET LIST OF GENERATED RANDOM VELOCITIES VX, VY, VZ '
      PRINT *, 'ENTER (Y OR N)'
      READ '(A)', ANS
      IF (ANS .EQ. 'Y') THEN
         PRINT *, 'ENTER THE NO. OF RANDOM VELOCITIES TO PRINT '
         PRINT *, 'UP TO NRVELOC=',NRVELOC
         READ  *, NOUT
* ....
         WRITE (6,27) NOUT,DELTAT
         DO 225 J=1,NOUT
            WRITE (6,29) J,VX(J),VY(J),VZ(J)
```

```
 225      CONTINUE
       ENDIF
 27       FORMAT (//,10X,'NUMBER OF RANDOM NUMBERS TO PRINT,NOUT=',I6,/,
     &            10X,'TIME STEP TO GENERATE THE RANDOM VELOCITIES '
     &            ,'VX, VY, VZ, DELTAT=',E10.3,/,T28
     &            ,'VX',T48,'VY',T68,'VZ','    (MEANS ARE SUBTRACTED)')
 29       FORMAT (10X,I4,T20,E14.7,T40,E14.7,T60,E14.7)
* ....
* .... PLOT RANDOM VELOCITY TIME SERIES VS TIME ....
       PRINT *, 'TO USE SUBROUTINE PLTSTND TO PLOT THE GENERATED RANDOM '
       PRINT *, 'VELOCITY VS TIME , ENTER (Y OR N) '
       READ '(A)', ANS
 882   IF (ANS .EQ. 'Y' ) THEN
          PRINT *, 'SELECT THE RANDOM VELOCITY TIME SERIES.  ENTER '
          PRINT *, ' VX OR VY OR VZ. '
          READ '(A)', ANS1
          PRINT *, 'ENTER THE LENGTH OF RANDOM VELOCITY TIME SERIES '
          PRINT *, ',LVPLT FOR PLOTTING UP TO NRVELOC =',NRVELOC
          READ  *, LVPLT
          IF ( ANS1 .EQ. 'VX' ) THEN
             DO 230 I=1,LVPLT
                Y(I)=VX(I)
 230         CONTINUE
          ELSEIF ( ANS1 .EQ. 'VY' ) THEN
             DO 232 I=1,LVPLT
                Y(I)=VY(I)
 232         CONTINUE
          ELSEIF ( ANS1 .EQ. 'VZ' ) THEN
             DO 234 I=1,LVPLT
                Y(I)=VZ(I)
 234         CONTINUE
          ENDIF
* ....
          CALL PLTSTND (Y,LVPLT,DELTAT,ANS1)
* ....
          PRINT *
          PRINT *
          PRINT *, 'DO YOU WANT TO PLOT ANY OTHER RANDOM VELOCITY '
          PRINT *, 'TIME SERIES ?  ENTER (Y OR N)'
          READ '(A)', ANS
          GO TO 882
       ENDIF
* .... EVALUATE PROBABILITY DISRIBUTION OF RANDOM VELOCITY ....
       PRINT *, 'DO YOU WANT TO EVALUATE PROBABILITY DISTRIBUTIONS OF'
       PRINT *, 'THE GENERATED RANDOM VELOCITIES ?  ENTER (Y OR N)'
       READ '(A)', ANS
       IF (ANS .EQ. 'Y') THEN
          DO 250 KPROB =1,3
             IF (KPROB .EQ. 1) THEN
                DO  240 I=1,NRVELOC
                   Y(I)=VX(I)
```

```
240           CONTINUE
              VARIANC=VXVAR
              ANS1 = 'VX'
           ELSEIF (KPROB .EQ. 2) THEN
              DO 242 I=1,NRVELOC
                 Y(I)=VY(I)
242           CONTINUE
              VARIANC=VYVAR
              ANS1 = 'VY'
           ELSEIF (KPROB .EQ. 3 ) THEN
              DO 244 I=1,NRVELOC
                 Y(I)=VZ(I)
244           CONTINUE
              VARIANC=VZVAR
              ANS1 = 'VZ'
           ENDIF
* .....    GENERATE UNITY VARIANCE RANDOM VELOCITY TIME SERIES    ....
           DO 245 I=1,NRVELOC
              Y(I)=Y(I)/SQRT(VARIANC)
245        CONTINUE
* ....
           CALL PROB (Y,NRVELOC,NBINS,VRANGE,PROBDIS)
* ....
           NBX2=2*NBINS
           NBX2M1=NBX2-1
           DELTAV=VRANGE/(NBINS-1)
           DO 246 I=1,NBX2M1
              X(I)=-VRANGE+(I-1)*DELTAV
246        CONTINUE
           WRITE (6,30) ANS1,NRVELOC,ANS1
           DO 248 I=1,NBX2
              IF (I .EQ. 1) THEN
                 WRITE (6,32) X(I),PROBDIS(I)
              ELSEIF (I .EQ. NBX2) THEN
                 IM1=I-1
                 WRITE (6,34) X(IM1),PROBDIS(I)
              ELSE
                 IM1=I-1
                 PROBDEN = PROBDIS(I)/DELTAV
                 XAVE    = (X(I)+X(IM1))/2.
                 STNDEN  = EXP(-0.5*XAVE**2)/SQRT(2.*PI)
                 WRITE (6,36) X(IM1),X(I),PROBDIS(I),XAVE,PROBDEN
     &                         ,STNDEN
              ENDIF
248        CONTINUE
250     CONTINUE
      ENDIF
30    FORMAT (//,10X,'PROBABILITY DISTRIBUTION OF RANDOM VELOCITY '
     &        ,'TIME SERIES',A4,' OF LENGTH = ',I5,//,10X
     &        ,'PROBABILITY OF VARIATES',15X,'MID-INTERVAL',5X
     &        ,'PROBABILITY DENSITY',5X,'STANDARD NORMAL',/,10X
```

```
     &       ,'IN THE INTERVAL',T66,'ORDINATES OF ',A4,T93
     &       ,'ORDINATES',/)
32    FORMAT (10X,'LESS THAN ',3X,'(',F6.2,') =',E11.4)
34    FORMAT (10X,'GREATER THAN ','(',F6.2,') =',E11.4)
36    FORMAT (15X,'(',F6.2,' ,',F6.2,') =',E11.4,T52,F6.2
     &       ,T68,E12.6,T90,E12.6)
* ....
* .... GENERATE FREQUENCY SPECTUM OF THE GENERATED RANDOM VELOCITY ....
* ....
      PRINT *, 'DO YOU WANT TO GENERATE THE FREQUENCY SPECTRUM OF '
      PRINT *, 'THE TIME SERIES ? ENTER (Y OR N)'
      READ '(A)', ANS
884   IF (ANS .EQ. 'Y') THEN
         PRINT *, 'INPUT ONE TIME SERIES TO GENERATE SPECTRUM. '
         PRINT *, 'ENTER VX OR VY OR VZ '
         READ '(A)', ANS1
         IF ( ANS1 .EQ. 'VX' )     THEN
            DO 252 I=1,NRVELOC
               Y(I)=VX(I)
252         CONTINUE
         ELSEIF ( ANS1 .EQ. 'VY' ) THEN
            DO 254 I=1,NRVELOC
               Y(I)=VY(I)
254         CONTINUE
         ELSEIF ( ANS1 .EQ. 'VZ' ) THEN
            DO 256 I=1,NRVELOC
               Y(I)=VZ(I)
256         CONTINUE
         ENDIF

* ....   LENGTH OF THE TIME SERIES HAS TO BE EVENLY DIVISIBLE     ....
* ....   BY L, LENGTH OF EACH SUBSEGMENT.  IF THIS CONDITION      ....
* ....   DEOS NOT MEET PAD BOTH TIME SERIES WITH ZEROES AT        ....
* ....   RIGHT END.                                               ....
* ....   NOTE: IF THE NUMBER OF GENERATED RANDOM VELOCITY         ....
* ....         COMPONENTS, NRVELOC, IS NOT EVENLY DIVISIBLE BY    ....
* ....         LENGTH OF THE SPECTRUM, LSPECT, THEN NRVELOC       ....
* ....         MUST BE SMALLER THAN THE DECLARED SIZE OF RANDOM   ....
* ....         VELOCITY COMPONENT ARRAYS AT MOST BY LSPECT.       ....
* ....                                                            ....
         LD2=LSPECT/2
         LD2P1=LD2+1
         NSEG=INT(NRVELOC/LSPECT)
         RNSEG=REAL(NRVELOC)/REAL(LSPECT)
         DIFF=RNSEG-NSEG
         IF(DIFF .NE. 0.0) THEN
            LTS=(NSEG+1)*LSPECT
            IPAD=NRVELOC+1
            DO 300 J=IPAD,LTS
               Y(J)=0.0
300         CONTINUE
```

```
      ENDIF
* ....
      CALL PSD (Y,LTS,LSPECT,LP2,DELTAT,PSY,ZY)
* ....
* ....  FORM THE FREQUENCY VECTOR   ....
* ....
      DO 325 I=1,LD2P1
         II=I-1
         F(I)=II/(LSPECT*DELTAT)
 325  CONTINUE
* ....
* ....  PRINT POWER SPECTRUM   ....
      WRITE (6,40) ANS1
      DO 340 I=1,LD2P1
         WRITE (6,42) F(I),PSY(I)
 340  CONTINUE
* ....  PRINT SUM OF THE POWER SPECTRA       ....
      SUMY=0.0
      DO 345 K=1,LD2P1
         SUMY=SUMY+PSY(K)
 345  CONTINUE
      WRITE (6,44) ANS1 , SUMY
 40   FORMAT(//,5X,'FREQUENCY',T20,'POWER SPECTRUM',/,T24,A4)
 42   FORMAT(4X,F10.4,T20,E14.7)
 44   FORMAT(//,10X,'SUM OF THE PSD OF(',A4,' )S=',E14.7)
* ....
* ....  ELIMINATE ZERO FREQUENCY FOR LOG-LOG PLOTTING   ....
* ....
      DO 370 I=2,LD2P1
         J=I-1
         F(J)=F(I)
 370  CONTINUE
* ....
* .... PLOT LOG-LOG SPECRTAL DENSITY OF RANDOM VELOCITY VS FREQUENCY ....
      PRINT *, 'TO USE PLTLOG TO PLOT THE SPECTRUM ENTER (Y OR N)'
      READ '(A)', ANS
      IF (ANS .EQ. 'Y')  THEN
* ....     GENERATE SPECTRUM VECTORS       ....
         DO 380 I=2,LD2P1
            J=I-1
            SOUT(J)=PSY(I)
 380     CONTINUE
* ....
* ....     PLOT POWER SPECTRUM       ....
* ....
         CALL PLTLOG (SOUT,F,LD2,LABEL,ANS1)
* ....
         PRINT *
      ENDIF

      PRINT *, 'DO YOU WANT SPECTRUM FOR OTHER TIME SERIES? '
```

```
   PRINT *, 'ENTER (Y OR N)'
   READ '(A)', ANS
   GO TO 884
ENDIF

PRINT *, 'DO YOU WANT TO PROCESS ANOTHER DATA FILE ? '
PRINT *, 'ENTER (Y OR N)'
READ '(A)', ANS
IF (ANS .EQ. 'Y') GO TO 881
STOP
END
```

```
      SUBROUTINE TURBS (XX,YY,ZZ,DELTAT,BEGINR,FINR,NPTS,PSI)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C     SUBROUTINE TURBS CONSTRUTS TURBULENCE VELOCITY COMPONENTS      C
C     ALONG THE BLADE FOR EACH AZIMUTH ANGLE AT EACH TIME STEP.      C
C     THE NUMBER OF POINTS ALONG THE BLADE AT WHICH TURBULENCE       C
C     VELOCITY IS EVALUATED IS GIVEN AS A PARAMETER , NPTS IN        C
C     PROGRAM SIMULX, AND CAN EASILY BE CHANGED.  THE TURBULENCE     C
C     VELOCITY COMPONENTS ARE COMPUTED AT EQUALLY DISTANCED          C
C     POINTS ALONG THE BLADE FROM AN INITIAL RADIUS TO A FINAL       C
C     RADIUS WHICH USER CAN DETERMINE.                               C
C     IN THE PRESENT ANALYSIS ONLY ONE RADIAL POSITION  AT THE       C
C     TIP WAS CONSIDERED (NPTS = 1).                                 C
C                                                                    C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INTEGER CONST,NPTS,NWCOMP
      REAL BEGINR,FINR,DELTAT,PSI,DIVIDER
      PARAMETER ( NWCOMP=12 )
      REAL XX(NPTS),YY(NPTS),ZZ(NPTS),U(NWCOMP),W(NWCOMP)
      REAL A(NWCOMP),B(NWCOMP),CC(NWCOMP),DD(NWCOMP)
      DOUBLE PRECISION SEED
      COMMON /TURBINE/ OMEGA,OMEGAZ,ROTR
      COMMON /WIND/ TL,TI,SW,VW
      COMMON /ATMOS/ A,B
      COMMON /RAND/ CONST, SEED, DIVIDER
      SAVE W
      DATA W /NWCOMP * 0.0/
* .... GENERATE COEFFICIENTS FOR FILTERS .....
      DO 10 I=1,NWCOMP
         AT=DELTAT*A(I)
         CC(I)=EXP(-AT)
         DD(I)=B(I)*SQRT((6.*SW/A(I))*(1.-EXP(-2.*AT)))
 10   CONTINUE
* .... GENERATE NWCOMP RANDOM NUMBERS .....
* ....
      CALL RANDOM (U,NWCOMP)
* ....
* .... GENERATE WIND VELOCITY COMPONENTS ....
      DO 20 I=1,NWCOMP
         U(I)=U(I)-0.5
         W(I)=CC(I)*W(I)+DD(I)*U(I)
 20   CONTINUE
      IF (NPTS .EQ. 1) THEN
         RSTEP=0.0
```

```
      ELSE
         RSTEP=(FINR-BEGR)/(NPTS-1)
      ENDIF
      R=BEGINR
      PSIX2=2*PSI
      ROTRSQ=ROTR*ROTR
      DO 30 I=1,NPTS
         R=R+(I-1)*RSTEP
         RSQ=R*R
         XX(I)=W(1)-(W(6)-W(7))*R*COS(PSI)
     &              -(W(8)-W(9))*R*SIN(PSI)
         YY(I)=W(2)+W(5)*R*COS(PSI)+W(4)*R*SIN(PSI)
     &              +W(10)*(RSQ-ROTRSQ/2.)
     &              +W(11)*RSQ*COS(PSIX2)+W(12)*RSQ*SIN(PSIX2)
         ZZ(I)=W(3)+(W(6)+W(7))*R*SIN(PSI)
     &              +(W(8)+W(9))*R*COS(PSI)
30    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ATMOS

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C  SUBROUTINE ATMOS COMPUTES THE TURBULENCE MODEL PARAMETERS A, B,     C
C  AND SW, WHERE A(I) AND B(I) ARE THE DIAGONAL ELEMENTS FOR           C
C  THE MATRICES IN THE WIND STATE EQUATION                             C
C  DX/DT = -A * X  +  B * W       AND W IS WHITE NOISE WITH PSD=SW.    C
C  THE EQUATIONS WERE DETERMINED BY LEAST SQUARE REGRESSION TO         C
C  DATA PRODUCED BY NUMERICAL COMPUTATION. (SEE REPORT)                C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INTEGER NWCOMP
      PARAMETER (NWCOMP=12)
      REAL ROTR,TI,TL,SW,VW
      REAL A(NWCOMP),B(NWCOMP)
      COMMON /TURBINE/ OMEGA,OMEGAZ,ROTR
      COMMON /WIND/ TL,TI,SW,VW
      COMMON /ATMOS/ A,B

* .... CALCULATE THE POWER SPECTRUM FOR THE NOISE INPUT ....

      SW=TL*(TI*TI)/VW/10000.
      RR=ROTR/TL
      VWSQ=VW*VW
      ROTRSQ=ROTR**2
      DIMCOA= VW/TL
      DIMCOBZ=VWSQ/TL
      DIMCOB1=VWSQ/(ROTR*TL)
      DIMCOB2=VWSQ/(ROTRSQ*TL)
      A(1)= (2.-2.894*RR*(1.-.1383*RR)/(1.+2.049*RR)) *DIMCOA
      B(1)= (2.-3.290*RR*(1.+.0270*RR)/(1.+2.054*RR)) *DIMCOBZ
      A(2)= (1.-1.713*RR*(1.-.0791*RR)/(1.+2.048*RR)) *DIMCOA
      B(2)= (SQRT(2.)-2.713*RR*(1.+.0159*RR)/(1.+2.051*RR))
     +   *DIMCOBZ
      A(3)= A(1)
      B(3)= B(1)
      A(4)= (.327/RR + .595 - .114*RR) * DIMCOA
      B(4)= (.281/RR**.25 + .645 - .150*RR) *DIMCOB1
      A(5)= A(4)
      B(5)= B(4)
      A(6)= (.434/RR + .917 - .153*RR) *DIMCOA
      B(6)= (.258/RR**.25 + .647 - .1093*RR) *DIMCOB1
      A(7)= (.5342/RR + 1.276 -.2147*RR) *DIMCOA
      B(7)= (.1167/RR**.25 + .7733 -.1284*RR) *DIMCOB1
```

```
A(8)= A(7)
B(8)= B(7)
A(9)= (1.654/RR + 1.069 + 2.154*RR) *DIMCOA
B(9)= (.3546/RR**.25 + .3951 + .2593*RR) *DIMCOB1
A(10)= (1.091/RR + .0276 + .0686*RR) *DIMCOA
B(10)= (.5508/RR**.25 + .6473 -.1365*RR) *DIMCOB2
A(11)= (1.081/RR + .0279 + .0685*RR) *DIMCOA
B(11)= (.3896/RR**.25 + .4567 -.0948*RR) *DIMCOB2
A(12)= A(11)
B(12)= B(11)
RETURN
END
```

```
      SUBROUTINE RANDOM (S,N)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                               C
C                                                               C
C    SUBROUTINE RANDOM GENERATES UNIFORMLY DITRIBUTED RANDOM     C
C    NUMBERS BETWEEN ZERO AND ONE USING POWER RESIDUE METHOD.   C
C                                                               C
C                                                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INTEGER CONST,N
      REAL DIVIDER,S(N)
      DOUBLE PRECISION SEED,INTPROD
      COMMON /RAND/ CONST, SEED, DIVIDER
      DO 10 I=1,N
         INTPROD=CONST*SEED
         IF (INTPROD .LT. DIVIDER) THEN
            S(I)=INTPROD
         ELSE
            S(I)=INTPROD-INT(INTPROD/DIVIDER)*DIVIDER
         ENDIF
         SEED=S(I)
         S(I)=S(I)/DIVIDER
 10   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MEANVAR (MEAN,VAR,S,N)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C                                                     C
C    SUBROUTINE MEANVAR COMPUTES MEAN AND VARIANCE    C
C    OF TIME SERIES.                                  C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INTEGER N
      REAL  MEAN,VAR,S(N)
      SUM=0.
      DO 20 I=1,N
         SUM=SUM+S(I)
 20   CONTINUE
      MEAN=SUM/FLOAT(N)
      DIFF=0.
      DO 30 I=1,N
         DIFF=DIFF+(S(I)-MEAN)**2
 30   CONTINUE
      VAR=DIFF/FLOAT(N)
      RETURN
      END
```

```
      SUBROUTINE PROB (VTS,LVTS,NBINS,VRANGE,PROBDIS)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C                                                     C
C    SUBROUTINE PROB COMPUTES PROBABILITY DISTIBUTION C
C    OF TIME SERIES.                                  C
C                                                     C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      INTEGER LVTS,NBINS,BINNUM
      REAL DELTAV,VRANGE
      REAL VTS(LVTS),PROBDIS(2*NBINS)
      NBX2=2*NBINS
      DO 20 I=1,NBX2
         PROBDIS(I)=0.0
 20   CONTINUE
      DELTAV=VRANGE/(NBINS-1)
      DO 30 I=1,LVTS
         IF ( VTS(I) .LT. 0.0 )  THEN
            IF ( VTS(I) .GE. -VRANGE )  THEN
               BINNUM=NBINS+INT(VTS(I)/DELTAV)
            ELSE
               BINNUM=1
            ENDIF
            PROBDIS(BINNUM)=PROBDIS(BINNUM)+1./LVTS
         ELSEIF ( VTS(I) .GE. 0.0 )  THEN
            IF ( VTS(I) .LE. VRANGE )  THEN
               BINNUM=NBINS+INT(VTS(I)/DELTAV)+1
            ELSE
               BINNUM=2*NBINS
            ENDIF
            PROBDIS(BINNUM)=PROBDIS(BINNUM)+1./LVTS
         ENDIF
 30   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PSD (Y,N,L,LP2,DT,PSY,ZY)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                  C
C   SUBROUTINE PSD USES FFT TO ESTIMATE THE FREQUENCY SPECTRUM OF  C
C   TIME SERIES                                                    C
C                                                                  C
C   ARGUMENTS                                                      C
C            Y     -INPUT VECTOR OF LENGTH N CONTAINING            C
C                   THE TIME SERIES.                               C
C            N     -INPUT LENGTH OF THE TIME SERIES.               C
C            L     -LENGTH OF THE TIME SERIES IN EACH SEGMENT.     C
C                   L MUST BE A POWER OF 2.                        C
C            LP2   -L=2**LP2 (L AS POWER OF TWO)                   C
C            LD2P1 -SPECTRAL COMPUTATIONS ARE AT                   C
C                   LD2P1= (L/2)+1 FREQUENCES.                     C
C            DT    -SAMPLING INTERVAL (SEC)                        C
C            PSY   -OUTPUT VECTOR OF LENGTH LD2P1 CONTAINING       C
C                   THE SPECTRAL ESTIMATES OF Y                    C
C                   NOTE THAT THE SPECTRAL ESTIMATES ARE           C
C                   TAKEN AT FREQUENCES (I-1)/(L*DT) (HERTZ)       C
C                   FOR I=1,2, ...,LD2P1                           C
C            ZY    -COMPLEX WORK VECTOR OF LENGTH L                C
C                                                                  C
C   REMARKS :                                                      C
C            1) THE SPECTRAL DENSITY FUNCTION IS DEFINED           C
C               ACCORDING TO EQ. 2.3 FROM CHAPTER TWO.             C
C            2) PRIOR TO CALLING PSD, THE MEAN OF TIME             C
C               SERIES Y SHOULD BE REMOVED FROM EACH               C
C               ELEMENT OF THE TIME SERIES.                        C
C            3) THE OUTPUT IS RETURNED IN UNITS WHICH ARE          C
C               THE (SQUARE OF THE DATA)/FREQUENCE                 C
C                                                                  C
C   SEGMENT AVERAGING IS USED TO OBTAIN THE SMOOTH ESTIMATES       C
C   THE TOTAL SAMPLE SIZE N = NSEG*L = NSEG*(2**LP2)               C
C   WHERE NSEG = NUMBER OF SEGMENTS                                C
C                                                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      REAL Y(N),PSY(L/2+1)
      COMPLEX ZY(L)


      LD2P1 = L/2 + 1
      NSEG  = INT(N/L)
      PI    = ACOS(-1.0)
```

```
C     SCALE FACTOR 0.875 IS DUE TO THE COSINE TAPPERING
C     TO ADJUST THE POWER SPECTRAL ESTIMATE RESULTS

      FACTOR=(DT*REAL(L))/(0.875)

C         INITIALIZE THE PSY

          DO 5 J=1,LD2P1
             PSY(J)=0.0
5         CONTINUE
          DO 50 I=1,NSEG
             ND=(I-1)*L
             DO 10 J=1,L
                JPND=J+ND
                ZY(J)=CMPLX(Y(JPND),0.0)
10           CONTINUE
* ....
* ....       TAPERING THE DATA SEQUENCE USING          ....
* ....       THE COSINE TAPER DATA WINDOW              ....
* ....
             CALL TAPER(ZY,L,DT)
* ....
* ....       COMPUTE DFT     ....
* ....
             CALL FFT(ZY,LP2,L)
* ....
             DO 30 J=1,LD2P1
                PSY(J)=PSY(J)+FACTOR*ABS(ZY(J))*ABS(ZY(J))
30           CONTINUE
50        CONTINUE

* ....    AVERAGE THE RESULTS FROM NSEG SEPARATE SEGMENTS    ....

          DO 60 I=1,LD2P1
             PSY(I)=PSY(I)/REAL(NSEG)
60           CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TAPER(ZY,L,DT)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                          C
C                                                          C
C     A SMOOTH FILTER SHAPE FOR FFT ESTIMATES TO           C
C     REDUCE LEAKAGE CAN BE OBTAINED BY TAPERING           C
C     THE ORIGINAL RANDOM TIME SERIES AT EACH END.         C
C     SUBROUTINE TAPER USES A COSINE TAPER DATA            C
C     WINDOW TO SMOOTH THE DATA AT 1/10 OF EACH            C
C     END OF THE RECORD (SEE FIG 11.8, PG 146, NEWLAND,    C
C     REFERENCE 1 IN APPENDIX B).                          C
C                                                          C
C     ARGUMENTS                                            C
C               ZY  -INPUT COMPLEX VECTOR OF LENNGTH       C
C                    L CONTAINING THE ORIGINAL DISCRETE    C
C                    TIME SERIES                           C
C                   -OUTPUT COMPLEX VECTOR OF LENGTH       C
C                    L CONTAING THE TAPERED DATA           C
C               L   -INPUT LENGTH OF THE TIME SERIES       C
C               DT  -SAMPLING INTERVAL                     C
C                                                          C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      COMPLEX ZY(L)

      PI=ACOS(-1.0)
      T=DT*REAL(L)
      TD10=T/10.0
      C1=9.0*TD10
      CONST=PI/TD10
      DO 20 I=1,L
         TIME=DT*REAL(I-1)
         IF (TIME .LE. TD10) THEN
             WT    = 0.5 - 0.5 * COS(CONST * TIME)
             ZY(I) = ZY(I)*WT
         ELSEIF (TIME .GE. C1) THEN
             WT    = 0.5 + 0.5 * COS(CONST * (TIME-C1))
             ZY(I) = ZY(I) * WT
         END IF
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FFT(A,NP,N)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                           C
C                                                           C
C     SUBROUTINE FFT FROM NEWLAND (PG 220), REFERENCE 1     C
C     APPENDIX B, CALCULATES THE DFT OF A SEQUENCE A(1),    C
C     A(2), ...,A(N), WHERE N = 2**NP, BY THE FFT METHOD.   C
C                                                           C
C     ARGUMENTS                                             C
C                A  -INPUT COMPLEX VECTOR OF LENGTH N       C
C                    CONTAINING THE DISCRETE TIME SERIES    C
C                   -OUTPUT COMPLEX VECTOR OF LENGTH N      C
C                    CONTAINING THE REQUIRED DFT            C
C                NP -N=2**NP                                C
C                N  -INPUT LENGTH OF THE TIME SERIES        C
C                                                           C
C                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      COMPLEX A(N),U,W,T


      PI=ACOS(-1.0)

C     DIVIDE ALL ELEMENTS BY N

      DO 1 J=1,N
         A(J)=A(J)/N
1        CONTINUE
      ND2=N/2
      NM1=N-1
      J=1
      DO 4 L=1,NM1
         IF (L .GE. J) GO TO 2
         T=A(J)
         A(J)=A(L)
         A(L)=T
2        K=ND2
3        IF (K .GE. J) GO TO 4
         J=J-K
         K=K/2
         GO TO 3
4        J=J+K
      DO 6 M=1,NP
         U=(1.0,0.0)
         ME=2**M
```

```
      K=ME/2
      W=CMPLX(COS(PI/K),-SIN(PI/K))
      DO 6 J=1,K
         DO 5 L=J,N,ME
            LPK=L+K
            T=A(LPK)*U
            A(LPK)=A(L)-T
5           A(L)=A(L)+T
6        U=U*W
      RETURN
      END
```

```
      SUBROUTINE PLTSTND (VTS,LVPLT,DELTAT,ANS1)
      INTEGER MARK,ICODE,IRATE,MODEL
      PARAMETER (MARK=0)
      REAL DELTAT,WIDTH,HEIGHT,VBIAS,TBIAS
      REAL VTS(LVPLT)
      CHARACTER *40 TIMELBL, VELCLBL, ANS1*2
      DATA ICODE/ 1 /,IRATE/ 2400 /,MODEL/ 4014 /
      DATA WIDTH/ 9.0 /,HEIGHT/ 7.0 /
      DATA TORIG/ 0.0 /, VORIG/ 0.0 /, TBIAS/ 3. /, VBIAS/ 1. /
      TIMELBL = 'TIME (SEC)'
      VELCLBL = 'RANDOM TURBULENCE VELOCITY '//ANS1
* .... FORM MIN & MAX ON THE TIME AXIS ....
                TMIN=0.0
                TMAX=LVPLT*DELTAT
                TFACT=WIDTH/(TMAX-TMIN)
* .... FIND MIN & MAX OF RANDOM VELOCITY VECTOR, VTS ....
                CALL CHECK (VTS,LVPLT,VMIN,VMAX)
                VFACT=HEIGHT/(VMAX-VMIN)
* ....
                CALL PLOTYPE (ICODE)
                CALL TKTYPE  (MODEL)
                CALL BAUD    (IRATE)
                CALL SIZE  (WIDTH+6. , HEIGHT+3.)
                CALL TEKPAUS
* ....
                CALL SCALE (TFACT,VFACT,TBIAS,VBIAS,TMIN,VMIN)
                CALL AXISL (TMIN,TMAX,TORIG,VMIN,VMAX,VORIG,0.0,1.0,
     &                      0 , 0 , -1 , 2 , 1. , 1. ,0.2, 0 )
* ..... PRINT HEADINGS ....
                XPOS=TMAX+.1/TFACT
                YPOS=-0.2/VFACT
                CALL SYMBOL (XPOS,YPOS,0.0,0.2,40,TIMELBL)
                XPOS=TMIN+1./TFACT
                YPOS=VMAX+0.4/VFACT
                CALL SYMBOL (XPOS,YPOS,0.0,0.2,40,VELCLBL)
* .... PLOT RANDOM VELOCITY ....
                CALL VECTORS
                IP=0
                DO 10 I=1,LVPLT
                   IJ=I-1
                   XT=IJ*DELTAT
                   YV=VTS(I)
                   CALL PLOT (XT,YV,IP,MARK)
                   IP=1
 10             CONTINUE
      CALL PLOTEND
      RETURN
      END
```

```
      SUBROUTINE PLTLOG (SPECT,FREQ,LHALF,LABEL,ANSPLT)
      INTEGER MARK,NLABEL,ICODE,IRATE,MODEL
      PARAMETER (NLABEL=1,MARK=26)
      REAL WIDTH,HEIGHT,FBIAS,SBIAS
      REAL SPECT(LHALF),FREQ(LHALF)
      CHARACTER *40  LABEL(NLABEL), FREQLBL, PSDLBL*60, ANSPLT*2
      DATA  ICODE/1/ IRATE/2400/ MODEL/4014/ WIDTH/9./ HEIGHT/7./
      DATA FBIAS/1./ SBIAS/1./
      FREQLBL = 'FREQ (HZ) '
      PSDLBL  = 'PSD OF '//ANSPLT//' '//LABEL(NLABEL)
* .... FIND MIN AND MAX OF THE FREQUENCY VECTOR ....
               CALL CHECK (FREQ,LHALF,FMINC,FMAXC)
               FMIN=ALOG10(FMINC)
               FMAX=ALOG10(FMAXC)
               FFACT=WIDTH/(FMAX-FMIN)
* .... FIND MIN & MAX OF THE SPECTRUM VECTOR ....
               CALL CHECK (SPECT,LHALF,SMINC,SMAXC)
               CALL RANGEL (SMINC,SMAXC,SMINR,SMAXR)
               SMIN=ALOG10(SMINR)
               SMAX=ALOG10(SMAXR)
               SFACT=HEIGHT/(SMAX-SMIN)
* ....
               CALL PLOTYPE(ICODE)
               CALL TKTYPE(MODEL)
               CALL BAUD(IRATE)
               CALL SIZE(WIDTH+2.5,HEIGHT+2.5)
               CALL TEKPAUS
* ....
               CALL SCALE (FFACT,SFACT,FBIAS,SBIAS,FMIN,SMIN)
               CALL AXISL (FMINC,FMAXC,FMINC,SMINC,SMAXC,SMINC,1.,1.
     &                     ,0,0,1,1,1.,1.,0.1,3)
* ....
* .... PRINT HEADINGS ....
* ....
               XPOS=FMIN+3.5/FFACT
               YPOS=SMIN-0.25/SFACT
               CALL SYMBOL (XPOS,YPOS,0.,0.2,40,FREQLBL)
               XPOS=FMIN+1./FFACT
               YPOS=SMAX+0.2/SFACT
               CALL SYMBOL (XPOS,YPOS,0.,0.2,60,PSDLBL)
* ....
* .... PLOT POWER SPECTRUM ....
* ....
               CALL POINTS
               IP=0
               DO 100 I=1,LHALF
                  XF=ALOG10(FREQ(I))
                  YS=ALOG10(SPECT(I))
                  CALL PLOT (XF,YS,IP,MARK)
                  IP=1
```

```
100          CONTINUE
      CALL PLOTEND
      RETURN
      END
```

APPENDIX D. INPUT DATA FILE

The following sample input data file is for Mod-0A turbine.

```
ROTATIONALLY SAMPELLED
 $INDATA CONST   = 16807          ,
         DELTAT  = 0.2            ,
         DIVIDER = 2147483647.    ,
         SEED    = 123457.        ,
         NRVELOC = 6300           ,
         OMEGA   = 40.            ,
         OMEGAZ  = 90.            ,
         ROTR    = 62.5           ,
         RRATIO  = 1.             ,
         TI      = 10.            ,
         TL      = 400.           ,
         VRANGE  = 3.             ,
         VW      = 17.9           ,
 $END
```

APPENDIX E. PROCEDURAL EXAMPLE OF THE PROGRAM SIMULX INTERACTIVE RUN

```
 ENTER NAME OF THE NEW DATA FILE
? DATMZAR

 ENTER THE NAME OF OUTPUT FILE
? OUTMZAR

      CONST    =   16807
      DELTAT   =        .200          (SEC)
      DIVIDER  =  .2147483647000E+10
      SEED     =  .1234570000000E+06
      NRVELOC  =    6300
      OMEGA    =      40.000          (RPM)
      OMEGAZ   =      90.000          (DEG)
      ROTR     =      62.500          (FEET)
      RRATIO   =       1.000
      TI       =      10.000          (PERCENT)
      TL       =     400.000          (FEET)
      VRANGE   =       3.000
      VW       =      17.900          (MILES/HR)

 DO YOU WANT TO CHANGE ANY VALUES ? ENTER(Y OR N)
? N
 TO GET LIST OF GENERATED RANDOM VELOCITIES UX, VY, VZ
 ENTER (Y OR N)
? N
 TO USE SUBROUTINE PLTSTND TO PLOT THE GENERATED RANDOM
 VELOCITY VS TIME , ENTER (Y OR N)
? N
 DO YOU WANT TO EVALUATE PROBABILITY DISTRIBUTIONS OF
 THE GENERATED RANDOM VELOCITIES ?  ENTER (Y OR N)
? Y
 DO YOU WANT TO GENERATE THE FREQUENCY SPECTRUM OF
 THE TIME SERIES ? ENTER (Y OR N)
? Y
 INPUT ONE TIME SERIES TO GENERATE SPECTRUM.
 ENTER UX OR VY OR VZ
? UX
 TO USE PLTLOG TO PLOT THE SPECTRUM ENTER (Y OR N)
? N
 DO YOU WANT SPECTRUM FOR OTHER TIME SERIES?
 ENTER (Y OR N)
? Y
 INPUT ONE TIME SERIES TO GENERATE SPECTRUM.
 ENTER UX OR VY OR VZ
? VY
 TO USE PLTLOG TO PLOT THE SPECTRUM ENTER (Y OR N)
? N
 DO YOU WANT SPECTRUM FOR OTHER TIME SERIES?
 ENTER (Y OR N)
? Y
 INPUT ONE TIME SERIES TO GENERATE SPECTRUM.
 ENTER UX OR VY OR VZ
? VZ
 TO USE PLTLOG TO PLOT THE SPECTRUM ENTER (Y OR N)
? N
 DO YOU WANT SPECTRUM FOR OTHER TIME SERIES?
 ENTER (Y OR N)
? N
 DO YOU WANT TO PROCESS ANOTHER DATA FILE ?
 ENTER (Y OR N)
? N
   42.224 CP SECONDS EXECUTION TIME.
/
```

APPENDIX F. RESULTS OF THE SAMPLE RUN FOR Mod-0A TURBINE

The simulated results are as observed from the tip of a Mod-0A wind turbine blade.

```
POWER RESIDUE METHOD WITH THE FOLLOWING PARAMETERS
IS USED TO GENERATE UNIFORMLY DISTRIBUTED RANDOM NUMBERS

     CONSTANT COEFF, CONST    =     16807
     SEED                     =     .12345700E+06
     MODULE DIVIDER, DIVIDER  =     .2147483647000E+10

              ACOEFF           BCOEFF

      1      .100271E+00      .277279E+01
      2      .524894E-01      .188229E+01
      3      .100271E+00      .277279E+01
      4      .175240E+00      .294581E-01
      5      .175240E+00      .294581E-01
      6      .240920E+00      .286800E-01
      7      .305939E+00      .258837E-01
      8      .305939E+00      .258837E-01
      9      .787020E+00      .275591E-01
     10      .460793E+00      .662569E-03
     11      .456611E+00      .468268E-03
     12      .456611E+00      .468268E-03

     NUMBER OF RANDOM VELOCITIES GENERATED, NRVELOC        =  6300
     TIME STEP TO GENERATE THE RANDOM VELOCITY, DELTAT     =   .20000E+00
     RADIAL DISTANCE TO SELECTED POINT ALONG THE ROTOR, R  =   .62500E+02
     ROTOR SPEED, OMGA                  =   .41888E+01     (RAD/SEC)
     INITIAL ROTATION, OMEGA-ZERO       =   .15708E+01     (RAD)
     WIND VELOCITY, VW                  =   .26253E+02     (FEET/SEC)
     TURBULENCE INTEGRAL SCALE, TL      =   .40000E+03     (FEET)
     TURBULENCE INTENSITY, TI           =   .10000E+02     (PERCENT)
     SPECTRUM OF THE INPUT WHITE NOISE, SU =               =   .15236E+00     (SEC)
     MEAN VALUE OF VX =  .2787366E+00    VARIANCE OF VX =  .6918183E+01
     MEAN VALUE OF VY =  .4602148E+00    VARIANCE OF VY =  .7846571E+01
     MEAN VALUE OF VZ = -.2425260E-01    VARIANCE OF VZ =  .5868340E+01

     PROBABILITY DISTRIBUTION OF RANDOM VELOCITY TIME SERIES  VX OF LENGTH =  6300

     PROBABILITY OF VARIATES            MID-INTERVAL     PROBABILITY DENSITY     STANDARD NORMAL
     IN THE INTERVAL                                     ORDINATES OF   VX          ORDINATES
```

LESS THAN	(-3.00) = .1587E-03			
	(-3.00 , -2.80) = .1111E-02	-2.90	.555556E-02	.595253E-02
	(-2.80 , -2.60) = .2063E-02	-2.70	.103175E-01	.104209E-01
	(-2.60 , -2.40) = .4444E-02	-2.50	.222222E-01	.175283E-01
	(-2.40 , -2.20) = .5397E-02	-2.30	.269841E-01	.283270E-01
	(-2.20 , -2.00) = .1063E-01	-2.10	.531746E-01	.439836E-01
	(-2.00 , -1.80) = .1476E-01	-1.90	.738095E-01	.656158E-01
	(-1.80 , -1.60) = .1921E-01	-1.70	.960317E-01	.940491E-01
	(-1.60 , -1.40) = .2651E-01	-1.50	.132540E+00	.129518E+00
	(-1.40 , -1.20) = .4000E-01	-1.30	.200000E+00	.171369E+00
	(-1.20 , -1.00) = .4778E-01	-1.10	.238889E+00	.217852E+00
	(-1.00 , -.80) = .4905E-01	-.90	.245238E+00	.266085E+00
	(-.80 , -.60) = .5889E-01	-.70	.294444E+00	.312254E+00
	(-.60 , -.40) = .6238E-01	-.50	.311905E+00	.352065E+00
	(-.40 , -.20) = .7175E-01	-.30	.358730E+00	.381388E+00
	(-.20 , .00) = .7333E-01	-.10	.366667E+00	.396953E+00
	(.00 , .20) = .7587E-01	.10	.379365E+00	.396953E+00
	(.20 , .40) = .7270E-01	.30	.363492E+00	.381388E+00
	(.40 , .60) = .7714E-01	.50	.385714E+00	.352065E+00
	(.60 , .80) = .6714E-01	.70	.335714E+00	.312254E+00
	(.80 , 1.00) = .6222E-01	.90	.311111E+00	.266085E+00
	(1.00 , 1.20) = .4873E-01	1.10	.243651E+00	.217852E+00
	(1.20 , 1.40) = .3825E-01	1.30	.191270E+00	.171369E+00
	(1.40 , 1.60) = .2317E-01	1.50	.115873E+00	.129518E+00
	(1.60 , 1.80) = .1810E-01	1.70	.904762E-01	.940491E-01
	(1.80 , 2.00) = .7619E-02	1.90	.380952E-01	.656158E-01
	(2.00 , 2.20) = .6825E-02	2.10	.341270E-01	.439836E-01
	(2.20 , 2.40) = .5556E-02	2.30	.277778E-01	.283270E-01
	(2.40 , 2.60) = .3492E-02	2.50	.174603E-01	.175283E-01
	(2.60 , 2.80) = .3968E-02	2.70	.198413E-01	.104209E-01
	(2.80 , 3.00) = .1111E-02	2.90	.555556E-02	.595253E-02
GREATER THAN	(3.00) = .6349E-03			

PROBABILITY DISTRIBUTION OF RANDOM VELOCITY TIME SERIES UV OF LENGTH = 6300

PROBABILITY OF VARIATES IN THE INTERVAL		MID-INTERVAL	PROBABILITY DENSITY ORDINATES OF UV	STANDARD NORMAL ORDINATES
LESS THAN	(-3.00) = .1111E-02			
	(-3.00 , -2.80) = .7937E-03	-2.90	.396825E-02	.595253E-02
	(-2.80 , -2.60) = .1587E-02	-2.70	.793651E-02	.104209E-01
	(-2.60 , -2.40) = .3492E-02	-2.50	.174603E-01	.175283E-01
	(-2.40 , -2.20) = .3968E-02	-2.30	.198413E-01	.283270E-01
	(-2.20 , -2.00) = .7302E-02	-2.10	.365079E-01	.439836E-01
	(-2.00 , -1.80) = .1381E-01	-1.90	.690476E-01	.656158E-01
	(-1.80 , -1.60) = .1889E-01	-1.70	.944444E-01	.940491E-01
	(-1.60 , -1.40) = .2730E-01	-1.50	.136508E+00	.129518E+00
	(-1.40 , -1.20) = .3635E-01	-1.30	.181746E+00	.171369E+00
	(-1.20 , -1.00) = .4683E-01	-1.10	.234127E+00	.217852E+00
	(-1.00 , -.80) = .5286E-01	-.90	.264286E+00	.266085E+00
	(-.80 , -.60) = .6254E-01	-.70	.312698E+00	.312254E+00
	(-.60 , -.40) = .7270E-01	-.50	.363492E+00	.352065E+00
	(-.40 , -.20) = .8444E-01	-.30	.422222E+00	.381388E+00
	(-.20 , .00) = .7683E-01	-.10	.384127E+00	.396953E+00
	(.00 , .20) = .7651E-01	.10	.382540E+00	.396953E+00

(.00 , .20) = .7651E-01	.10	.382540E+00	.396953E+00
(.20 , .40) = .7381E-01	.30	.369048E+00	.381388E+00
(.40 , .60) = .6889E-01	.50	.344444E+00	.352065E+00
(.60 , .80) = .5635E-01	.70	.281746E+00	.312254E+00
(.80 , 1.00) = .4841E-01	.90	.242063E+00	.266085E+00
(1.00 , 1.20) = .4206E-01	1.10	.210317E+00	.217852E+00
(1.20 , 1.40) = .3222E-01	1.30	.161111E+00	.171369E+00
(1.40 , 1.60) = .3063E-01	1.50	.153175E+00	.129518E+00
(1.60 , 1.80) = .2016E-01	1.70	.100794E+00	.940491E-01
(1.80 , 2.00) = .1365E-01	1.90	.682540E-01	.656158E-01
(2.00 , 2.20) = .1302E-01	2.10	.650794E-01	.439836E-01
(2.20 , 2.40) = .6667E-02	2.30	.333333E-01	.283270E-01
(2.40 , 2.60) = .4286E-02	2.50	.214286E-01	.175283E-01
(2.60 , 2.80) = .9524E-03	2.70	.476190E-02	.104209E-01
(2.80 , 3.00) = .9524E-03	2.90	.476190E-02	.595253E-02
GREATER THAN (3.00) = .6349E-03			

PROBABILITY DISTRIBUTION OF RANDOM VELOCITY TIME SERIES VZ OF LENGTH = 6300

PROBABILITY OF VARIATES IN THE INTERVAL	MID-INTERVAL	PROBABILITY DENSITY ORDINATES OF VZ	STANDARD NORMAL ORDINATES
LESS THAN (-3.00) = .3175E-03			
(-3.00 , -2.80) = .4762E-03	-2.90	.238095E-02	.595253E-02
(-2.80 , -2.60) = .2381E-02	-2.70	.119048E-01	.104209E-01
(-2.60 , -2.40) = .3016E-02	-2.50	.150794E-01	.175283E-01
(-2.40 , -2.20) = .3333E-02	-2.30	.166667E-01	.283270E-01
(-2.20 , -2.00) = .9048E-02	-2.10	.452381E-01	.439836E-01
(-2.00 , -1.80) = .1111E-01	-1.90	.555556E-01	.656158E-01
(-1.80 , -1.60) = .2048E-01	-1.70	.102381E+00	.940491E-01
(-1.60 , -1.40) = .2619E-01	-1.50	.130952E+00	.129518E+00
(-1.40 , -1.20) = .4032E-01	-1.30	.201587E+00	.171369E+00
(-1.20 , -1.00) = .4746E-01	-1.10	.237302E+00	.217852E+00
(-1.00 , -.80) = .5778E-01	-.90	.288889E+00	.266085E+00
(-.80 , -.60) = .6651E-01	-.70	.332540E+00	.312254E+00
(-.60 , -.40) = .6841E-01	-.50	.342063E+00	.352065E+00
(-.40 , -.20) = .7254E-01	-.30	.362698E+00	.381388E+00
(-.20 , .00) = .7762E-01	-.10	.388095E+00	.396953E+00
(.00 , .20) = .7302E-01	.10	.365079E+00	.396953E+00
(.20 , .40) = .7365E-01	.30	.368254E+00	.381388E+00
(.40 , .60) = .6302E-01	.50	.315079E+00	.352065E+00
(.60 , .80) = .6175E-01	.70	.308730E+00	.312254E+00
(.80 , 1.00) = .5778E-01	.90	.288889E+00	.266085E+00
(1.00 , 1.20) = .4349E-01	1.10	.217460E+00	.217852E+00
(1.20 , 1.40) = .3857E-01	1.30	.192857E+00	.171369E+00
(1.40 , 1.60) = .2524E-01	1.50	.126190E+00	.129518E+00
(1.60 , 1.80) = .2048E-01	1.70	.102381E+00	.940491E-01
(1.80 , 2.00) = .1587E-01	1.90	.793651E-01	.656158E-01
(2.00 , 2.20) = .7619E-02	2.10	.380952E-01	.439836E-01
(2.20 , 2.40) = .5556E-02	2.30	.277778E-01	.283270E-01
(2.40 , 2.60) = .2540E-02	2.50	.126984E-01	.175283E-01
(2.60 , 2.80) = .2381E-02	2.70	.119048E-01	.104209E-01
(2.80 , 3.00) = .6349E-03	2.90	.317460E-02	.595253E-02
GREATER THAN (3.00) = .1429E-02			

FREQUENCY	POWER SPECTRUM UX
.0000	.6049776E+02
.0391	.1963951E+02
.0781	.9142553E+01
.1172	.4392374E+01
.1563	.1679976E+01
.1953	.1372716E+01
.2344	.6355281E+00
.2734	.4387497E+00
.3125	.4920447E+00
.3516	.3293057E+00
.3906	.4551208E+00
.4297	.3742490E+00
.4688	.3353939E+00
.5078	.4426219E+00
.5469	.8901370E+00
.5859	.1336677E+01
.6250	.2356935E+01
.6641	.3142170E+01
.7031	.2461201E+01
.7422	.9707069E+00
.7813	.9767400E+00
.8203	.4996871E+00
.8594	.2980353E+00
.8984	.2608651E+00
.9375	.1700205E+00
.9766	.1378442E+00
1.0156	.1494149E+00
1.0547	.1313543E+00
1.0938	.1022334E+00
1.1328	.6868052E-01
1.1719	.7944809E-01
1.2109	.5845696E-01
1.2500	.6064474E-01
1.2891	.5818579E-01
1.3281	.4364089E-01
1.3672	.5203108E-01
1.4063	.3601760E-01
1.4453	.4075913E-01
1.4844	.3749977E-01
1.5234	.3325957E-01
1.5625	.3163256E-01
1.6016	.3018275E-01
1.6406	.2819096E-01
1.6797	.3315463E-01
1.7188	.2854833E-01
1.7578	.2549400E-01
1.7969	.2379286E-01
1.8359	.2121934E-01
1.8750	.3290871E-01
1.9141	.2368969E-01
1.9531	.2438889E-01
1.9922	.2110549E-01
2.0313	.2244903E-01
2.0703	.3337699E-01
2.1094	.1591815E-01
2.1484	.2182798E-01
2.1875	.2470305E-01
2.2266	.2160245E-01
2.2656	.1546142E-01
2.3047	.2449096E-01
2.3438	.2279807E-01
2.3828	.1826092E-01
2.4219	.1863145E-01
2.4609	.2582036E-01
2.5000	.2748395E-01

SUM OF THE PSD OF(UX)S= .1152977E+03

FREQUENCY	POWER SPECTRUM UY
.0000	.8989386E+02
.0391	.1706530E+02
.0781	.4204839E+01
.1172	.1871292E+01
.1563	.1153121E+01
.1953	.1167095E+01
.2344	.6118766E+00
.2734	.3259570E+00
.3125	.3297577E+00
.3516	.3092288E+00
.3906	.2687522E+00
.4297	.3199178E+00
.4688	.3419386E+00
.5078	.3487520E+00
.5469	.5458201E+00
.5859	.9514736E+00
.6250	.2168570E+01
.6641	.5168488E+01
.7031	.3560563E+01
.7422	.1244913E+01
.7813	.7072755E+00
.8203	.4480872E+00
.8594	.2789543E+00
.8984	.2527932E+00
.9375	.2019094E+00
.9766	.1031491E+00
1.0156	.1340067E+00
1.0547	.1318198E+00
1.0938	.1766182E+00
1.1328	.2443295E+00
1.1719	.2587490E+00
1.2109	.4152425E+00
1.2500	.7203783E+00
1.2891	.9657761E+00
1.3281	.1071430E+01
1.3672	.8244152E+00
1.4063	.5735425E+00
1.4453	.3685053E+00
1.4844	.3094007E+00
1.5234	.2187015E+00
1.5625	.1403191E+00

1.5625	.1403191E+00
1.6016	.1014524E+00
1.6406	.6037793E-01
1.6797	.8760638E-01
1.7188	.6238084E-01
1.7578	.6009705E-01
1.7969	.6577859E-01
1.8359	.4889308E-01
1.8750	.3500600E-01
1.9141	.3819243E-01
1.9531	.3244359E-01
1.9922	.3436401E-01
2.0313	.3036546E-01
2.0703	.3361854E-01
2.1094	.3162102E-01
2.1484	.3414904E-01
2.1875	.2999253E-01
2.2266	.2434398E-01
2.2656	.2874651E-01
2.3047	.2303529E-01
2.3438	.2394717E-01
2.3828	.2398075E-01
2.4219	.2571705E-01
2.4609	.3092459E-01
2.5000	.2949189E-01

SUM OF THE PSD OF(VY)S= .1413842E+03

FREQUENCY	POWER SPECTRUM VZ
.0000	.4910821E+02
.0391	.2016754E+02
.0781	.5251017E+01
.1172	.2365756E+01
.1563	.1620577E+01
.1953	.9069718E+00
.2344	.6585728E+00
.2734	.5539867E+00
.3125	.4445584E+00
.3516	.3142210E+00
.3906	.4005998E+00
.4297	.3685486E+00
.4688	.3684053E+00
.5078	.4969580E+00
.5469	.6630746E+00
.5859	.1274137E+01
.6250	.2100307E+01
.6641	.3123941E+01
.7031	.2578653E+01
.7422	.1575120E+01
.7813	.6152442E+00
.8203	.3963874E+00
.8594	.4205281E+00
.8984	.2710570E+00
.9375	.1969253E+00
.9766	.1866573E+00
1.0156	.1274903E+00
1.0547	.1000479E+00
1.0938	.9053876E-01
1.1328	.7616853E-01
1.1719	.7246598E-01
1.2109	.6363103E-01
1.2500	.7237054E-01
1.2891	.5166605E-01
1.3281	.6642795E-01
1.3672	.4371515E-01
1.4063	.4235130E-01
1.4453	.4652180E-01
1.4844	.4171425E-01
1.5234	.3293063E-01
1.5625	.2695288E-01
1.6016	.2829352E-01
1.6406	.2749536E-01
1.6797	.3116947E-01
1.7188	.2850512E-01
1.7578	.3514961E-01
1.7969	.2498404E-01
1.8359	.2847206E-01
1.8750	.2664573E-01
1.9141	.2991973E-01
1.9531	.2602818E-01
1.9922	.2111585E-01
2.0313	.2447086E-01
2.0703	.2554371E-01
2.1094	.2189080E-01
2.1484	.1749072E-01
2.1875	.2466256E-01
2.2266	.2096707E-01
2.2656	.2287831E-01
2.3047	.2143503E-01
2.3438	.1595903E-01
2.3828	.2233711E-01
2.4219	.2581138E-01
2.4609	.2196439E-01
2.5000	.2767664E-01

SUM OF THE PSD OF(VZ)S= .9798381E+02

END OF FILE

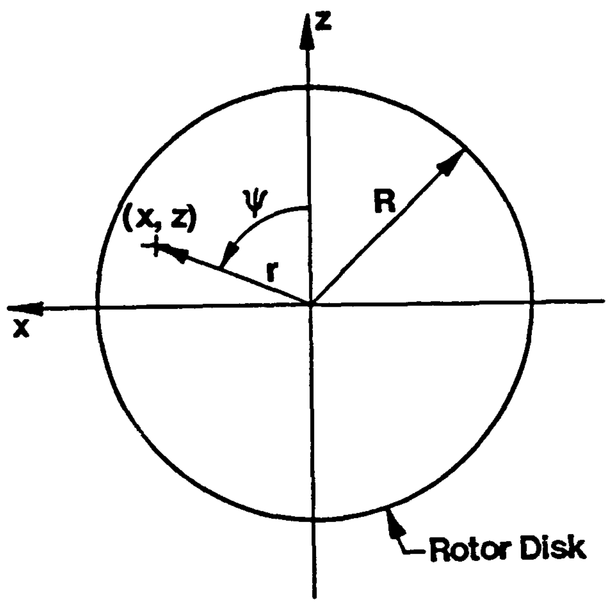

Figure 1.1. Rotor disk coordinate system.

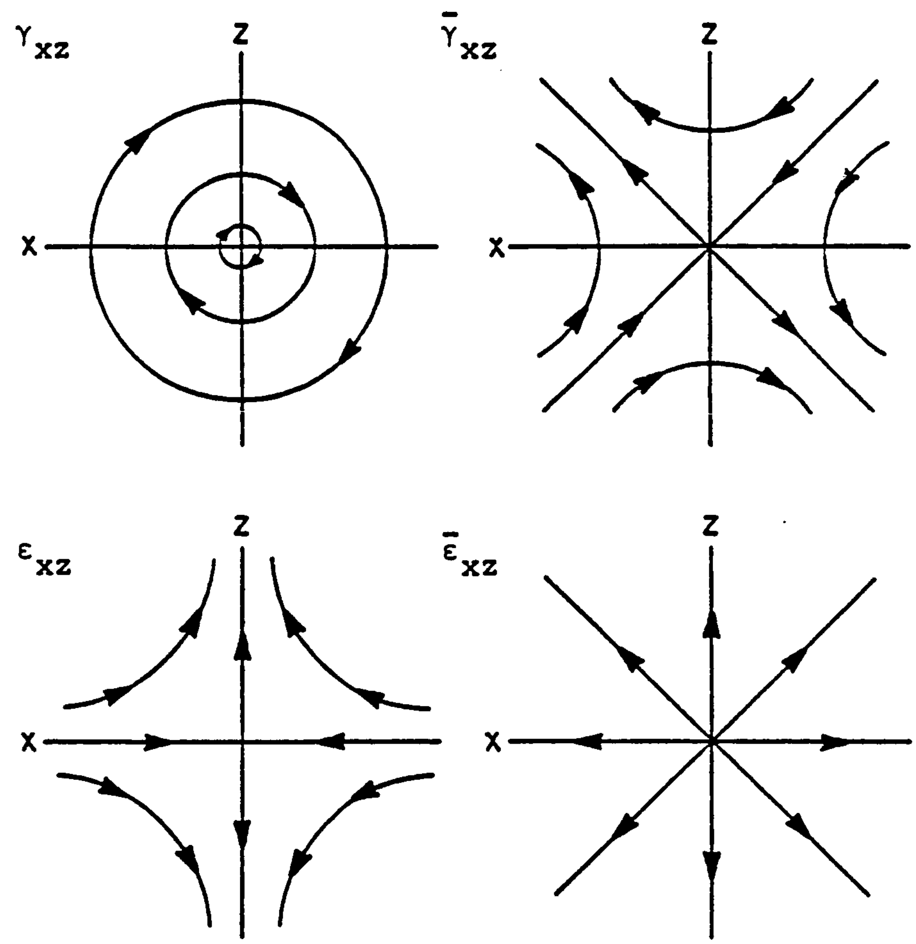

Figure 1.2. Streamlines for in-plane velocity gradient terms.